FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Reconhecimento de Elementos da Língua Gestual Portuguesa com Kinect

Miguel Medeiros Correia

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Eurico Manuel Elias de Morais Carrapatoso (PhD)

Co-orientador: António Abel Vieira de Castro (PhD)

22 de Julho de 2013

U.PORTO

FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

MIEEC - MESTRADO INTEGRADO EM ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES 2012/2013

A Dissertação intitulada

"Reconhecimento de Elementos da Língua Gestual Portuguesa com Kinect"

foi aprovada em provas realizadas em 22 Julho 2013

o júri

Presidente Professor Doutor Miguel Fernando Paiva Velhote Correia
Professor Auxiliar do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

Professora Doutora Ana Maria Perfeito Tomé
Professora Associada do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro

Professor Doutor Eurico Manuel Elias Morais Carrapatoso
Professor Auxiliar do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

Professor Doutor António Vieira de Castro
Professor Adjunto do ISEP-IPP

O autor declara que a presente dissertação (ou relatório de projeto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extratos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são corretamente citados.

Autor - Miguel Medeiros Correia

Faculdade de Engenharia da Universidade do Porto

Resumo

O reconhecimento de Língua Gestual é uma área de investigação relativamente recente. As soluções atuais dependem da interação de inúmeros sistemas, elevando assim a sua complexidade e custo.

Com este trabalho pretendeu-se provar que é possível simplificar estes sistemas. O objetivo principal estava no desenvolvimento de procedimentos simples que usam o sensor Kinect da Microsoft e as suas ferramentas de desenvolvimento para obter reconhecimento em tempo real. Como a deteção e seguimento do movimento humano são funcionalidades do sensor, este trabalho foca-se em dois elementos da Língua Gestual Portuguesa: a expressão facial e o gesto estático.

Para alcançar os objetivos propostos apresentaram-se duas aplicações, cada uma focada num dos elementos estudados. Para a expressão facial usa-se a informação disponibilizada pelo sensor e o pacote de desenvolvimento *Face Tracking SDK* para, através de parametrização, conseguir detetar e reconhecer expressões faciais utilizadas na Língua Gestual Portuguesa. Utilizou-se a máscara CANDIDE-3, disponível no SDK e as unidades de animação para parametrizar as expressões, utilizando como base o código FACS.

O reconhecimento de gestos estáticos foi de seguida estudado. Utiliza-se a informação de profundidade produzida pelo sensor para simplificar o estágio de pré-processamento, geralmente complexo neste tipo de aplicações. Utiliza-se um método simples, que usufrui da capacidade do Kinect de rastreio do esqueleto do utilizador, para conseguir detetar e segmentar a área de interesse da imagem onde se encontra a mão do utilizador.

Uma vez tendo a mão do utilizador segmentada passa-se ao processo de extração de características relevantes que ajudem na deteção de padrões que diferenciem cada gesto. Utilizam-se os 7 momentos de imagem invariantes de Hu. Estes dão-nos características do contorno do objeto úteis que são resistentes a alterações face a rotação, translação e alteração de escala. São, também, utilizados os ângulos das duas primeiras componentes principais do objeto que vem permitir distinguir ligeiras alterações na orientação do objeto.

Extraídas as características passa-se à classificação. Usam-se vetores de 9 características (7 momentos invariantes de Hu e os ângulos das duas componentes principais) para criar famílias de padrões que traduzem a morfologia de cada um dos gestos estudados.

Como os dois algoritmos utilizados precisam de ser treinados, apresenta-se uma interface capaz de levantar um série de amostras e classificá-las de forma a criar um conjunto de padrões de treino a serem usados pelos classificadores.

Finalmente, apresenta-se a interface da aplicação desenvolvida para o reconhecimento de gestos estáticos que permite deteção e reconhecimento em tempo real. Com um *dataset* de treino de 300 amostras e usando o classificador K-NN com $K = 5$, consegue-se obter uma taxa de viabilidade de 96.5%.

Com este trabalho prova-se que, usando o sensor Kinect e empregando métodos simples e de baixa complexidade consegue-se seguir e detetar o movimento Humano de forma viável a servir para aplicações de reconhecimento da Língua Gestual Portuguesa.

Abstract

Sign Language recognitions is a relatively new field of research. The current solutions rely on the interaction of several systems, increasing it's complexity and cost.

With this project we intended to prove that it is possible to simplify these systems. The main goal was the development of simple procedures that use Microsoft's Kinect sensor and it's development tools to achieve real time recognition. As detection and tracking of Human movement is one of the sensor's features, this work focuses on two elements of Portuguese Sign Language: facial expressions and static gestures.

To achieve the proposed objectives we present two applications, each focused on one of the studied elements. For the facial expression we use the information provided by the sensor and the development package *Face Tracking SDK* so that, through parameterization, we might detect and recognize facial expressions used in Portuguese Sign Language. The CANDIDE-3 mask and animation units available in the SDK, were used to model the expressions, using FACS coding as base.

The static gesture recognition was then investigated. The depth information produced by the sensor is used to simplify the pre-processing stage that is generally complex in this kind of applications. A simple method that takes advantage of Kinect's skeleton tracking ability is used to detect and segment the image's area of interest where the user's hand is located.

Once the hand is segmented the following step is to extract relevant features which help in detecting patterns that distinguish every gesture. We use the 7 Hu's invariant moments. These give us useful characteristics of the object's outline that are resistant to rotation, translation and scale changes. We also take advantage of the angles of the object's first two principal components that allow to distinguish slight changes on the object's orientation.

After feature extraction, the next step is classification. Vectors of 9 characteristics (7 Hu's invariant moments and 2 main component angles) are used to create families of patterns which reflect the morphology of each of the studied gestures.

Since the two algorithms used need to be trained, we introduce an interface able to create samples and classify them in order to generate a set of training patterns to be used by the classifiers.

Finally, we present the application interface developed for static gestures recognizing, which allows for real time gesture recognition. With a training dataset of 300 samples and using the K-NN classifier, making $K = 5$, we can achieve a recognition success rate of 96.5%.

With this work we prove that, using the Kinect sensor and employing simple and low complexity methods we can track and detect Human movement in such a way that it can be used in Portuguese Sign Language recognition applications.

Agradecimentos

Em primeiro lugar agradeço ao Professor Doutor Eurico Carrapatoso e Professor Doutor António Castro por terem acreditado em mim, neste tema e por terem aceite acompanhar-me no seu desenvolvimento. Um obrigado por todo o tempo, confiança, paciência, ajuda e motivação que me prestaram desde o primeiro dia.

Quero deixar um agradecimento aos Doutores Jaime S. Cardoso e Jorge Alves da Silva e à Psicóloga Ana Bela Baltazar pelo tempo que me disponibilizaram para discutir este tema. Conversas estas que me ajudaram a progredir com as melhores noções e maior conhecimento.

Também quero agradecer ao Eduardo Magalhães, assim como a todos os investigadores, professores e estudantes associados ao Laboratório de Música Eletrónica por me concederem um espaço para trabalhar assim como pela sua companhia e apoio durante todo o desenvolvimento deste projeto.

À minha família e em particular os meus pais, que sem eles não estaria aqui, obrigado por todo o apoio durante todos estes anos. Obrigado por estarem sempre disponíveis e pela paciência que sempre demonstraram comigo.

Deixo aqui um obrigado a todos os meus colegas e amigos que por estes anos cruzaram os seus caminhos com o meu. Os nomes são muitos para enumerar mas convosco aprendi muito e cresci. A vossa companhia durante este anos foi indispensável ao meu sucesso. Quero deixar um especial agradecimento ao meu colega e amigo Rui Costa, muitas batalhas travamos lado a lado e finalmente chegamos ao fim deste caminho. Obrigado pela tua amizade e por estares sempre ponto a lançar-te comigo em novos desafios.

Por último, um especial obrigado à minha namorada, Suzana Vale. Obrigado por me ouvires, aturares nos piores dias, pela tua compreensão e, acima de tudo, obrigado por acreditares em mim e me dares forças para continuar quando nem eu acreditava.

Miguel Correia

"Computers are like Old Testament gods;
lots of rules and no mercy."

Joseph Campbell

Conteúdo

1 Introdução **1**
- 1.1 Caracterização do tema . . . 2
- 1.2 Objetivos . . . 3
- 1.3 Motivação . . . 4
- 1.4 Estrutura do documento . . . 4

2 Estado da arte **7**
- 2.1 Deteção e rastreio . . . 8
  - 2.1.1 Deteção baseada no pixel . . . 8
    - 2.1.1.1 Desafios na modelação do plano de fundo . . . 9
    - 2.1.1.2 Modelação estatística do plano de fundo . . . 10
    - 2.1.1.3 Supressão de sombras . . . 12
  - 2.1.2 Deteção baseada no objeto . . . 13
    - 2.1.2.1 Conceito e desafios . . . 14
    - 2.1.2.2 Abordagens à deteção de objetos . . . 14
    - 2.1.2.3 Segmentação dos planos da imagem . . . 15
    - 2.1.2.4 Rastreio . . . 16
- 2.2 Língua Gestual . . . 16
  - 2.2.1 Língua Gestual Portuguesa . . . 17
  - 2.2.2 Aquisição e reconhecimento de dados . . . 18
  - 2.2.3 Características manuais . . . 18
    - 2.2.3.1 Forma da mão . . . 19
    - 2.2.3.2 Ortografia gestual . . . 19
  - 2.2.4 Características não manuais . . . 20
- 2.3 Sumário . . . 20

3 Ferramentas e arquitetura **23**
- 3.1 Microsoft Kinect . . . 23
  - 3.1.1 Aplicações desenvolvidas com Kinect . . . 24
  - 3.1.2 Sensor Kinect . . . 25
    - 3.1.2.1 Sensor de profundidade . . . 26
    - 3.1.2.2 Câmara RGB, motor, acelerómetro e microfones . . . 27
  - 3.1.3 Imagens de profundidade – RGB-D . . . 27
  - 3.1.4 Rastreio do esqueleto . . . 27
  - 3.1.5 Rastreio da posição da cabeça e da expressão facial . . . 28
- 3.2 Microsoft KinectSDK . . . 29
- 3.3 *OpenCV* e *Emgu CV* . . . 30
- 3.4 Arquitetura do sistema . . . 30

3.5 Sumário . . . 32

4 Desenvolvimento **35**
4.1 Reconhecimento da expressão facial . . . 35
4.1.1 Microsoft Face Tracking SDK . . . 35
4.1.2 CANDIDE-3 . . . 36
4.1.3 Unidades de animação . . . 38
4.1.4 Expressão facial na LGP e sua deteção . . . 40
4.2 Deteção e reconhecimento de gestos estáticos . . . 42
4.2.1 Pré-processamento . . . 43
4.2.1.1 Deteção da mão . . . 45
4.2.1.2 Segmentação . . . 47
4.2.2 Características da mão . . . 47
4.2.2.1 Contornos . . . 48
4.2.2.2 Momentos . . . 50
4.2.2.3 Momentos invariantes de Hu . . . 51
4.2.2.4 Análise de componentes principais . . . 52
4.2.3 Classificadores . . . 54
4.2.3.1 *K-Nearest Neighbours* . . . 55
4.2.3.2 *Support Vector Machine* . . . 56
4.2.4 Sistema de treino . . . 57
4.2.5 Ambiente de deteção . . . 59
4.3 Sumário . . . 60

5 Análise de resultados **63**
5.1 Reconhecimento da expressão facial . . . 63
5.1.1 Deteção em tempo real . . . 63
5.1.2 *Face Tracking SDK* . . . 64
5.1.3 Unidades de animação . . . 65
5.1.4 Apreciação global . . . 66
5.2 Reconhecimento de gestos estáticos . . . 66
5.2.1 Características . . . 67
5.2.2 Classificadores . . . 70
5.2.2.1 *K-Nearest Neighbours* . . . 72
5.2.2.2 *Support Vector Machine* . . . 74
5.2.3 Deteção em tempo real . . . 75
5.2.4 Apreciação global . . . 76
5.3 Sumário . . . 76

6 Conclusões **79**
6.1 Resultados . . . 79
6.2 Trabalho Futuro . . . 80

Referências **85**

Lista de Figuras

2.1 Geração de sombras sobre um objecto. Adaptado de [SMO03]. 12
2.2 Supressão de sombras [MHKS11]. 13
2.3 Alfabeto gestual usado na Língua Gestual Portuguesa. Adaptado de [dS11]. . . . 16

3.1 Componentes do sensor Kinect [Mic12]. 26
3.2 Processo de estimação da posição das articulações do corpo humano desenvolvido por Shotton *et al.* Adaptado de [SFC+11]. 28
3.3 À esquerda a imagem de profundidade obtida pelo Kinect e à direita o resultado correspondente do algoritmo desenvolvido por Cai *et al.* [CGZZ10]. 29
3.4 Diagramas de alto nível do sistema. 31

4.1 Pontos rastreados (13 não são apresentados) pelo *Face Tracking SDK* [MSD12]. . 36
4.2 Ângulos usados pelo *Face Tracking SDK* para traduzir a pose da cabeça do utilizador [MSD12]. 37
4.3 Pontos de deteção (a vermelho) usados pelo *Face Tracking SDK* projetados no espaço tridimensional. A azul mostra-se a máscara CANDIDE-3. 38
4.4 Interface desenvolvida para deteção da expressão facial. 41
4.5 Deteção da expressão facial, com o sistema desenvolvido. 43
4.6 Bits de um pixel de uma imagem de profundidade. Adaptado de [WA12]. 44
4.7 Captura da cena através do *Depth Stream*, pré-processamento e segmentação da mão. 45
4.8 Diagramas da classes implementadas que operam especificamente com a informação da mão. 46
4.9 Envelope convexo (linha vermelha) e defeitos de convexidade. 48
4.10 Ilustração do maior círculo que cabe na área convexa formada pelos defeitos relevantes. 49
4.11 Características da mão geradas em tempo real pela aplicação desenvolvida. . . . 50
4.12 Ilustração da análise de componentes principais. Adaptado de [Dav96]. 53
4.13 Classes de padrões escolhidas para o sistema de reconhecimento. 54
4.14 Diagramas da classes implementadas para supervisionar a operação dos classificadores K-NN e SVM. 55
4.15 Representação do principio de SVM. Adaptado de [Dav96]. 56
4.16 Interface do sistema de treino. 57
4.17 Classe de gestão de escrita e leitura de informação de características para um ficheiro XML. 58
4.18 Interface do sistema de deteção. 60

5.1 Deteção errada do estado da boca. 65

5.2 Rastreio das articulações do utilizador, note-se que devido à posição do braço o sistema não deteta bem a distância entre o cotovelo e o pulso. 67
5.3 Erros produzidos pela análise do envelope convexo para cada gesto considerado. . 68
5.4 Taxas de viabilidade para o classificador K-NN. O vetor de caraterísticas usado contem apenas os 7 momentos de Hu. 69
5.5 Efeito da variação do número de amostras/gesto no conjunto de treino e do valor de K no classificador K-NN. 70
5.6 Desempenho do classificador K-NN. Influência do valor de K, da normalização dos ângulos das componentes em diferentes intervalos e do número de amostras/gesto no conjunto de treino. 72
5.7 Comportamento do classificador K-NN para diferentes intervalos de normalização dos ângulos das 2 primeiras componentes principais. São usadas 300 amostras/gesto e $K = 5$. 73
5.8 Taxas de erro/letra para os melhores intervalos obtidos com K-NN. São usadas 300 amostras/gesto e $K = 5$. 74
5.9 Taxas de viabilidade para o classificador SVM com a variação da gama de normalização dos ângulos das componentes principais 74
5.10 Taxas de erro/letra observadas com o classificador SVM para diferentes intervalos de normalização dos ângulos das 2 primeiras componentes principais. Foram usadas 300 amostras/gesto. 75

Lista de Tabelas

4.1 Gama de valores de *pitch*, *roll* e *yaw* retornados pelo *Face Tracking SDK*. Adaptado de [MSD12]. 38
4.2 Unidades de animação rastreadas pelo *Face Tracking SDK*. Adaptado de [MSD12]. 39
4.3 Expressões faciais utilizadas na Língua Gestual Portuguesa. Adaptado de [Bal10]. 40
4.4 Relação entre as unidades de animação do sistema FACS com as disponíveis no *Face Tracking SDK*, para cada expressão facial implementada. 41

5.1 Valores demonstrativos dos momentos de Hu para cada classe de gestos. 70

Abreviaturas e Símbolos

AU	Animation Unit
ASL	American Sign Language
DMF	Deformable Model Fitting
DPM	Deformable Part-based Model
FACS	Facial Animation Coding System
fps	Frames per Second
GPU	Graphics Processing Unit
HOG	Histogram of Oriented Gradients
HSV	Hue, Saturation and Value
KDE	Kernel Distribution Estimation
K-NN	K-Nearest Neighbours
LGP	Língua Gestual Portuguesa
LP	Língua Portuguesa
ML	Machine Learning
MoG	Mixture of Gaussians
OpenCV	Open Source Computer Vision Library
PCA	Principal Component Analysis
PSO	Particle Swarm Optimization
RGB-D	Red, Green, Blue and Depth data
SDK	Software Development Kit
SLR	Sign Language Recognition
SOV	Sujeito-Objeto-Verbo
SU	Shape Unit
SVM	Support Vector Machine

Capítulo 1

Introdução

A linguagem é uma capacidade mental complexa que se desenvolve em criança de forma inconsciente e informal. Surge sem uma perceção da sua lógica subjacente. Por isto Pinker [Pin07], define-a como um instinto, diz-nos para olhar para a linguagem não como uma prova da singularidade humana mas como uma adaptação biológica para comunicar informação. Por sua vez, a língua é a materialização dessa capacidade usando um determinado conjunto de regras, um código, seja este realizado através da fala, gesto, imagem ou escrita.

A língua usada para comunicação depende do grupo de indivíduos que a usam. Podemos categorizar o tipo de comunicação em dois grupos, o oral e o não oral. No primeiro insere-se a língua falada, como a Língua Portuguesa, enquanto no segundo temos a escrita, o gesto e a imagem. Podemos, ainda, dividir em termos de *emissor-recetor*, categorizando conforme o método de emissão e o de receção. Para aqueles que ouvem, os ouvintes, a língua estabelece-se em termos *orais-auditivos* enquanto que para os não ouvintes, ou surdos, geralmente estabelece-se em termos *gestuais-visuais*, onde gestual define-se como o conjunto de elementos linguísticos manuais, corporais e faciais necessários para a articulação de um sinal.

Foi Darwin em 1871 na sua obra *A Descendência do Homem* [Dar71] o primeiro a articular a linguagem como uma espécie de instinto, dizendo primeiro que a língua é uma arte visto que tem que ser aprendida. No entanto, uma criança apresenta uma tendência instintiva para comunicar, como pode ser observado pelo seu balbucio. A língua é aprendida pelo contacto, por ouvir aqueles que nos rodeiam e por imitação. Esta forma de aprendizagem é muito difícil de um surdo usar uma vez que não tem acesso ao *feedback* auditivo. Torna-se então, para este, extremamente difícil produzir o som da palavra. É por isso que usam a Língua Gestual, uma modalidade de comunicação baseada no *gestual-visual*. É esta a sua língua materna, é esta a língua que usam diariamente para comunicar e na qual os seus pensamentos são formulados. A Língua Gestual que usam, em Portugal denominada Língua Gestual Portuguesa (LGP), tem uma estruturação, ou uma sintaxe, completamente diferente da empregue na Língua Portuguesa. Esta diferença sintática dificulta a compreensão total quando a comunicação é apresentada de uma forma escrita.

Na Língua Gestual, enquanto o emissor constrói uma oração a partir dos elementos manuais, corporais e faciais, o recetor usa o sistema percetual visual, ao invés do sistema percetual auditivo,

para entender o que é comunicado. Assim, a informação linguística é construída tendo em conta as capacidades percetuais do sistema visual humano. Desta forma, as relações espaciais na Língua Gestual são muito complexas.

Não é natural, para aqueles que usam a Língua Gestual, a produção ou compreensão de língua escrita uma vez que a sua língua natural possui uma estrutura paralela, com a utilização de gestos complexos que envolvem simultaneamente diversas partes do corpo do sinalizador. Desta forma, a aprendizagem da Língua Portuguesa pelos surdos é um processo de aquisição de uma segunda língua, o que acaba por dificultar a compreensão de texto escrito na estrutura da Língua Portuguesa.

1.1 Caracterização do tema

O desenvolvimento de tecnologias de comunicação como o rádio e a televisão sempre teve como alvo a pessoa ouvinte. Cada programa é feito na língua do país de origem e traduzido para a língua do público alvo, seja por dobragem ou por legendagem. No entanto, as duas soluções não cobrem as necessidades da pessoa surda.

No caso da legendagem, a estrutura não é a da língua principal deste, dificultando a compreensão completa da informação. A solução mais viável praticada é a utilização de intérpretes, pessoas que conseguem traduzir o que é dito para a sua forma gestual. Esta solução nem sempre é possível, seja pela dificuldade da interpretação em si, seja pelo custo de produção, ou até pelo atraso intrínseco gerado pela forma de comunicação.

A Língua Gestual é uma língua complexa. Tem a sua própria sintaxe e é composta por diversos elementos organizados no espaço. A complexidade do gesto realizado com as mãos, a posição e trajetória que estas fazem no espaço tridimensional, a expressão facial durante a execução do movimento e até o posicionamento do corpo são, todos em conjunto, o que trazem significado a um gesto.

A natureza multi-modal do gesto na língua, assim como a deteção dos ligeiros e precisos movimentos das mãos, são dos principais desafios no reconhecimento da Língua Gestual. Estes problemas levaram a que a maioria das soluções tecnológicas fossem complexas e baseadas nas mais variadas tecnologias, criando uma confusão de sistemas que mal colaboram entre si ou são demasiado ineficientes.

Enquanto que o reconhecimento automático de fala avançou já ao ponto de estar comercialmente disponível, o reconhecimento de Língua Gestual é uma área investigação muito recente. Existe uma necessidade de desenvolver métodos e mecanismos capazes de traduzir, de forma viável e acessível, a Língua Gestual. O indivíduo surdo não dispõe facilmente de ferramentas que lhe permitem traduzir corretamente a sua língua. O desenvolvimento desta área tem o benefício de trazer a esta comunidade as capacidades de interpretação entre línguas que temos hoje em dia com ferramentas de reconhecimento automático de fala. Poderá também disponibilizar uma forma de

interpretação da Língua Gestual mais acessível trazendo a interpretação a mais conteúdos multimédia, através da introdução da adequada tradução para a Língua Gestual da mesma forma que acontece com a legendagem.

Este trabalho foca-se em preencher a lacuna na falta de ferramentas eficientes e acessíveis de interpretação da Língua Gestual.

1.2 Objetivos

Pensamos que seria útil existirem formas viáveis de interpretação da Língua Gestual, para transmitir a informação produzida pelo gesto em tempo real e de uma forma de simples compreensão, intuitiva e natural.

Novos desenvolvimentos na área de visão por computador trazem-nos tecnologias que prometem resolver problemas desta categoria. Uma dessas tecnologias é o sensor *Kinect* da Microsoft, um dispositivo de interação baseado em sensores de movimentos criado inicialmente como interface de controlo da plataforma de jogos Xbox 360 e agora melhorado para ser utilizado com um computador Windows.

O Kinect oferece-nos uma forma fácil e rápida de rastrear o movimento de um sujeito. O movimento do tronco, da cabeça, dos membros e agora, com a última versão, obtém-se, também, um fácil rastreio facial. Um dos desafios é o rastreio de pequenos movimentos efetuados pelas mãos e a distinção entre estes.

O objetivo principal deste trabalho foi provar que com apenas o sensor Kinect e as ferramentas de desenvolvimento disponibilizadas pela Microsoft se conseguem criar métodos que cobrem todas as vertentes da Língua Gestual.

Como a deteção e rastreio dos movimentos do corpo humano são duas das características do sensor, considerou-se que o Kinect seria capaz de gerar suficiente informação para seguir o movimento que o gesto cria no espaço. Assim, o foco do trabalho está na criação de métodos que mostram que o sensor é também capaz de detetar a expressão facial e os elementos manuais da Língua Gestual em tempo real.

No que toca aos elementos manuais, estes podem ser divididos em duas categorias: gestos dinâmicos, ou seja com movimento ao longo do tempo, e gestos estáticos, como as letras do alfabeto gestual, ou ortografia gestual. Este trabalho foca-se na segunda categoria procurando, em primeiro lugar, provar que se consegue efetivamente detetar as pequenas variações que os dedos fazem para representar um gesto.

Assim, numa fase inicial, será importante estudar os paradigmas da deteção e rastreio do movimento humano. O estudo aprofundado destas técnicas ajudará na melhor compreensão das dificuldades e limitações que esta área de investigação apresenta. Permitirá também compreender quais as verdadeiras vantagens que o sensor Kinect proporciona no âmbito de visão por computador. É necessário também estudar a Língua Gestual em si. Iremos explorar os avanços na prática de reconhecimento da Língua Gestual a fim de criar soluções contextualizadas e eficientes.

Pretende-se recorrer a métodos que usem o sensor Kinect e o software de desenvolvimento da Microsoft capazes de detetar gestos manuais e a expressão do utilizador em tempo real, provando, assim, que este sensor se apresenta como uma tecnologia viável à criação de sistemas de reconhecimento e interpretação da Língua Gestual Portuguesa a baixo custo.

1.3 Motivação

A área de sistemas multimédia e a possibilidade da sua aplicação nas mais variadas áreas, desde a interação homem-computador, à análise de imagem e áudio sempre interessou pessoalmente ao autor. O gosto pela inovação e desenvolvimento tecnológico, aliado a um desejo pessoal de criar novas soluções práticas e viáveis a problemas ou necessidades comuns são um dos grandes motivadores. Problemas complexos podem sempre ser divididos em pequenos problemas exequíveis.

Os avanços na tecnologia de reconhecimento de fala e a atual capacidade de sintetizar e analisar voz em tempo real levaram a considerar o caso da Língua Gestual e a falta de mecanismos similares. Esta é uma área de investigação ainda num estágio muito inicial, novos algoritmos e novos sistemas de aquisição têm vindo a surgir. A possibilidade de contribuir numa área de investigação, trazer a uma comunidade um novo mecanismo de comunicação e o gosto pessoal do autor pela abordagem a problemas complexos são fatores que contribuíram fortemente à proposta deste projeto.

1.4 Estrutura do documento

Este documento é composto por 6 capítulos, sendo este o primeiro onde introduzimos o tema e caracterizámos o problema em mãos explicitando o seu contexto. Apresentámos os objetivos principais do projeto e a motivação por detrás da sua elaboração.

No capítulo 2 apresenta-se o estado da arte. Neste começamos por estudar as principais abordagens utilizadas na área de deteção e rastreio do movimento do corpo humano. Passamos então a estudar a Língua Gestual e os elementos que constituem um gesto, analisando abordagens ao reconhecimento de cada um destes.

De seguida, no capítulo 3, introduz-se o sensor Kinect, apresentando com algum pormenor o seu funcionamento e as suas características. Passa-se de seguida a um estudo das ferramentas que serviram de apoio ao desenvolvimento do projeto, finalizando com a arquitetura do sistema desenvolvido.

No capítulo 4 é apresentado todo o trabalho que levou à conceção das aplicações de reconhecimento da expressão facial e de gestos estáticos. Para ambos os sistemas, aborda-se todo o processo que leva à deteção e reconhecimento, evidenciando as características de cada segmento das aplicações que contribuem para o produto final.

Uma vez exposto todo o processo de desenvolvimento e o funcionamento interno das aplicações de deteção passa-se, no capítulo 5, a analisar o funcionamento destas. É feita uma análise de

todas as componentes dos sistemas de forma a avaliar os seus comportamentos face aos objetivos individuais.

Por fim, no capítulo 6, são apresentadas as conclusões finais ao projeto comentando os resultados obtidos. É, também, apresentada uma proposta de futuros trabalhos que visam melhorar ou aumentar as funcionalidades dos sistemas desenvolvidos.

Capítulo 2

Estado da arte

Para podermos detetar e identificar os elementos gestuais usados numa oração em Língua Gestual Portuguesa de uma forma automática, usando visão por computador, precisamos de efetivamente extrair a informação necessária de um sinal de vídeo. Não é difícil imaginar a amplitude do problema. Todos já assistimos, numa Língua ou noutra, a um intérprete ou um surdo a comunicar com Língua Gestual. A amplitude e complexidade dos movimentos, seja no espaço ou na própria variedade dos elementos gestuais, apresentam-se como obstáculos para a realização de um sistema eficiente e rápido.

Nas últimas décadas muita investigação tem aparecido na área de visão por computador. O surgimento de aparelhos e sensores, capazes de captar imagens e vídeo com maior e melhor resolução, a um preço acessível, assim como o aumento do poder de processamento das máquinas vieram abrir as portas para o desenvolvimento de mais e melhores técnicas na área de análise do movimento humano.

Neste capítulo vamos analisar a tecnologia que tem vindo a aparecer, focando-nos principalmente na área de deteção e análise do movimento humano. Começamos, primeiro, por estudar o primeiro passo em qualquer sistema de deteção, após a aquisição da imagem: a deteção em si. Na secção 2.1 analisamos as principais abordagens utilizadas para este problema. Analisamos como é efetuado o processo de separar as características relevantes numa imagem, separando o plano de fundo do cenário, ou *background*, que contem informação irrelevante ao contexto do problema, do primeiro plano (*foreground*). É neste último plano que se encontra a pessoa que queremos seguir e analisar. Estudamos os maiores problemas encontrados neste processo e como os ultrapassar.

Finalmente, na secção 2.2 focamo-nos com mais precisão na área de deteção e análise da Língua Gestual, começando por analisar questões linguísticas na Língua Gestual Portuguesa, passando então a estudar as tendências e investigações nesta área de franca evolução nas últimas décadas.

2.1 Deteção e rastreio

Quando o objetivo é o reconhecimento de pessoas ou objetos numa imagem, na área de visão por computador, o primeiro passo é o de reconhecer se na imagem em questão está ou não presente o objeto pretendido, assim como onde este se encontra. A esta tarefa é dada o nome de *deteção* [MHKS11], mais propriamente, o termo *figure-ground segmentation* é usado na terminologia inglesa e descreve o processo pelo qual o sistema visual organiza um cenário em figuras de primeiro plano (*foreground*) e fundo (*background*).

Processos de deteção são geralmente aplicados como o primeiro estágio de muitos sistemas de captura e análise sendo, portanto, um passo crucial. Existem, basicamente, duas abordagens ao problema de deteção: *deteção baseada no pixel* e *deteção baseada no objeto*. Na primeira abordagem, cada pixel de uma nova imagem é comparada com um modelo do cenário analisando se este pertence ao fundo ou ao primeiro plano. O resultado desta análise para todos os pixels da nova imagem retorna a silhueta de todas as pessoas detetadas. A deteção baseada no objeto tem como princípio movimentar uma janela deslizante por toda a imagem sendo calculada a probabilidade da existência de uma pessoa para cada posição da janela. Neste tipo de abordagem o resultado, geralmente, surge como uma caixa que engloba as pessoas detetadas na imagem. Estas duas abordagens são descritas em mais detalhe nas secções 2.1.1 e 2.1.2, respetivamente.

Em aplicações como deteção de intrusos é necessário uma série de imagens consecutivas para que seja possível qualquer processamento. Nestes casos o *rastreio* de objetos é um requisito. Na literatura, a noção de rastreio toma definições diferentes. Segundo Moeslund [MHK11] este é composto de dois processos: deteção e correspondência temporal, em que o último é definido como o processo de associar os objetos detetados na imagem atual com aqueles detetados nas imagens prévias, retornando assim trajetórias temporais no espaço.

2.1.1 Deteção baseada no pixel

Num vasto número de sistemas de análise de movimentos são usadas câmaras estacionárias para monitorizar atividade em cenários de exterior ou de interior. Como a câmara é estacionária a deteção pode ser alcançada por simplesmente comparar o plano de fundo com cada nova imagem. A esta técnica dá-se o nome de *subtração do plano de fundo*. Uma das suas maiores vantagens é o facto de retornar uma eficiente segmentação das regiões do primeiro plano e do fundo da imagem. Subtração de fundo é muito usada para processamento posterior, como rastreio, e foi-o desde os primeiros sistemas de análise de movimento humano, como o *Pfinder* [WADP96].

A noção de comparar cada pixel de uma imagem a um modelo do cenário onde esta se encontra é simples de se compreender. No entanto, esta abordagem depende do facto de considerar que o cenário é fixo. Tal consideração pode ser adaptada facilmente ao caso de uma imagem no interior, onde podemos controlar o ambiente e as condições de luz. Porém, o caso muda de figura quando consideramos cenários de exterior, onde as árvores mexem-se com o vento e as sombras movimentam-se com a posição do sol. Por esta razão, o desenvolvimento desta área tem vindo a focar-se em formas de modelar os pixels do plano de fundo e como atualizar estes modelos

durante o processamento. Nesta secção vamos analisar os maiores desafios na modelação de fundo, passando a discutir alguns dos métodos mais usados para a implementar subtração de fundo.

2.1.1.1 Desafios na modelação do plano de fundo

Para uma boa segmentação da imagem, é necessário ter uma boa modelação do fundo do cenário. Para o efeito é preciso fazer com que o modelo tolere alterações, seja tornando-o invariante a estas ou adaptativo. Toyama *et al.* [TKBM99] identificam uma lista de dez desafios que um modelo de plano de fundo tem de superar: *moved objects*, *time of day*, *light switch*, *waving trees*, *camouflage*, *bootstrapping*, *foreground aperture*, *sleeping person*, *waking person* e *shadows*. Por outro lado, Elgammal *et al.* [EDHD02] usam a origem da alteração para a classificar:

Alterações de Iluminação

Alterações de iluminação no cenário podem ocorrer como:

- Alterações graduais em cenários de exterior, devido ao movimento do sol relativamente ao cenário;
- Alterações repentinas como o ligar e desligar de um interruptor de luz num cenário de interior;
- Sombras projetadas por objetos no plano de fundo ou pelo movimento de objetos no primeiro plano.

Alterações de Movimento

Alterações de movimento podem ser categorizadas como:

- Deslocamento global da imagem por ligeiros desvios da câmara. Apesar de assumirmos que a câmara é estacionária pequenos deslocamentos desta podem ocorrer por fatores externos como a força do vento;
- Movimento dos elementos do plano de fundo, como o movimento das árvores com o vento.

Alterações Estruturais

Estas são alterações introduzidas ao plano de fundo da imagem pelos objetos alvo. Elgammal [MHKS11] define este tipo de alteração como algo que ocorre tipicamente quando qualquer objeto relativamente permanente é introduzido no plano de fundo do cenário. Como por exemplo, se uma pessoa se mantém estacionária no cenário por algum tempo. Toyama *et al.* [TKBM99] dividem esta categoria em *moved objects*, *sleeping person* e *waking person*.

Uma das questões centrais na modelação do plano de fundo é a decisão de que características modelar. Podem-se usar características baseadas no pixel como a intensidade ou bordas (estas podem ser identificadas como zonas na imagem, que apresentam variação local de intensidade significativa) ou características baseadas na região como o bloco da imagem. A escolha das características a modelar irá influenciar a tolerância do modelo a alterações.

Outra questão reside na escolha do modelo estatístico representativo das observações do sistema para cada pixel ou região. A escolha deste modelo irá afetar o grau de precisão da deteção. Na secção seguinte referem-se os métodos estatísticos mais utilizados no contexto de modelação do plano de fundo.

2.1.1.2 Modelação estatística do plano de fundo

Ao nível do pixel, podemos pensar no problema de subtração do plano de fundo como a necessidade de classificar se a intensidade de um determinado pixel, x_t, observada no instante t, pertence ao plano de fundo ou ao primeiro plano da imagem. No entanto, como a intensidade de um pixel do primeiro plano pode tomar qualquer valor arbitrário, podemos assumir que a sua distribuição é uniforme. Assim, reduzimos um problema de classificação de duas classes a um de apenas uma classe. Esta classificação pode ser obtida através do historial de observações que está disponível desse pixel.

Modelação paramétrica

A maioria das técnicas de subtração do plano de fundo, usa como base o modelo de plano de fundo de gaussiana única [MHKS11], segundo o qual, considerando que a distribuição de ruído de um determinado pixel tem uma distribuição gaussiana nula $N\left(0,\sigma^2\right)$, tem-se que a intensidade desse pixel é uma variável aleatória com distribuição gaussiana $N\left(\mu,\sigma^2\right)$.

A estimação dos parâmetros deste modelo reduz-se a avaliar o valor médio e a variância das observações das intensidades dos pixels, ao longo do tempo. Assim, a utilização deste modelo na prática reduz-se a subtrair uma imagem de fundo B a cada nova imagem I_t e verificar se a diferença é superior a um determinado limiar. Neste caso, a imagem de fundo B é composta pelo valor médio das imagens do plano de fundo.

Este modelo pode ser adaptado a variações lentas no cenário pela atualização iterativa da imagem de fundo. Uma solução eficiente é conhecida por *esquecimento exponencial* [MHKS11]:

$$B_t = \alpha I_t + (1-\alpha) B_{t-1} \tag{2.1}$$

onde $t \geq 1$, B_t representa a imagem de plano de fundo calculada até à imagem t e α corresponde à velocidade do esquecimento da informação do plano de fundo. Esta equação funciona como um filtro passa-baixo com ganho α que separa de uma forma eficaz o plano de fundo dos objetos em movimento. É de notar que neste caso B_t passa a representar a tendência central da imagem de fundo ao longo do tempo [GBCR00]. Este modelo é usado em sistemas como o *Pfinder* [WADP96].

Tipicamente, em cenários de exterior, a imagem de fundo não é completamente estática, podendo variar a intensidade de determinados pixels de imagem para imagem. Nestes casos a abordagem de um única gaussiana não retorna bons resultado. Friedman *et al.* [FR97]

apresentam um modelo onde uma mistura de três distribuições gaussianas foram usadas para modelar o valor dos pixels em aplicações de vigilância de tráfego, usando cada uma das distribuições para representar a estrada, veículos e sombras, respetivamente. Desde o trabalho de Friedman, melhoramentos ao modelo de *mistura de gaussianas* - em inglês *Mixture of Gaussians*, *MoG* - foram apresentados, como por exemplo, o trabalho de Stauffer e Grimson [GSRL98].

Modelação não paramétrica

Em cenários de exterior é habitual haver uma vasta gama de variações muito rápidas, como ondas do mar. Este tipo de variações fazem parte do plano de fundo e a modelação deste tipo de cenário requer uma representação mais flexível da distribuição de cada pixel [EHD00].

Uma técnica geral para estimar a função densidade de probabilidade de uma variável é a técnica de Estimação da Densidade do Núcleo - *KDE (Kernel Density Estimation)*. Os estimadores de núcleo convergem assintoticamente para qualquer função de densidade com amostras suficientes. Utilizando esta técnica pode-se evitar a necessidade de guardar o conjunto de dados completo, aplicando pesos a subconjuntos de amostras. Elgammal *et al.* [EHD00] introduzem uma abordagem para a modulação do plano de fundo, utilizando esta técnica. Seja $x_1, x_2, \ldots, x_N$ uma amostra de intensidades de um pixel, pode-se obter uma aproximação da função de densidade de probabilidade para a intensidade do pixel, a qualquer intensidade, como:

$$Pr(x_t) = \frac{1}{N} \sum_{i=1}^{N} \prod_{j=1}^{d} K_{\sigma_j} \left(x_{t_j} - x_{i_j} \right) \tag{2.2}$$

A função 2.2 encontra-se generalizada de forma a usar características de cor. Nesta, x_t é uma característica de cor de dimensão d num instante t e K_{σ_j} representa a função de núcleo com largura de banda σ_j na dimensão espacial de cor j [MHKS11].

Como função de núcleo podem ser usadas várias funções com propriedades diferentes, embora na literatura seja habitual o uso da função gaussiana. Neste caso, a função gaussiana é apenas usada para atribuir pesos. Ao contrário da modelação paramétrica, esta técnica é mais geral e não assume que a função densidade tenha qualquer forma específica.

Ao usar esta estimação de probabilidade, um pixel x_t é considerado como parte do primeiro plano se $Pr(x_t) < th$, onde th é um limiar associado globalmente que pode se ajustado conforme necessário.

Um dos problemas da utilização de técnicas KDE é a escolha de uma boa largura de banda do núcleo (σ). Teoricamente, quando o número de amostras tende para infinito a influência da largura de banda decresce tornando-se desprezável, mas na prática é usado um número finito de amostras. Uma largura de banda muito baixa dará lugar a uma estimação irregular, enquanto que um fator muito elevado conduzirá a uma estimação demasiado suavizada. Como são esperadas diferentes variações na intensidade de pixel de um local para o outro

da imagem, é usado uma largura de banda de núcleo diferente para cada pixel. Mittal e Paragios apresentaram uma abordagem adaptativa para a estimação da largura de banda do núcleo [MP04].

2.1.1.3 Supressão de sombras

Um processo de subtração do plano de fundo irá sempre detetar sombras de objetos como se fizessem parte do objeto em si. Quando os objetos são estáticos a sua sombra pode ser modelada juntamente com o plano de fundo. No entanto, a deteção de sombras de objetos que se movimentam apresenta um problema: as sombras confundem-se com o objeto a ser detetado. Pense-se no caso de análise do movimento humano, a existência da sombra do indivíduo dificulta a deteção correta do movimento dos membros.

Pode-se evitar detetar sombras ou até suprimir a sua deteção com a compreensão de como surgem. As sombras são constituídas por duas partes. Na figura 2.1 pode-se observar a representação da sombra de um objeto que se movimenta. A parte mais escura, a *umbra*, não recebe luz da fonte luminosa, e a parte mais clara, a *penumbra*, recebe alguma luz da fonte [SMO03]. Devido às condições de luz direta e indireta, típicas de cenários de interior e exterior, é comum encontrar sombras de penumbra. Este tipo de sombra pode ser caracterizada como tendo uma intensidade menor, preservando a crominância do plano de fundo [MHKS11].

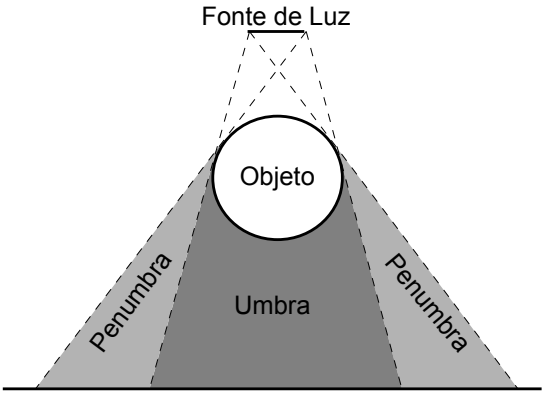


Figura 2.1: Geração de sombras sobre um objecto. Adaptado de [SMO03].

Devido a esta propriedade de invariância à crominância, geralmente são utilizados espaços de cor também invariantes, ou menos sensíveis a alterações de intensidade de cor. Este é o caso do sistema HSV (*Hue, Saturation and Value*), onde as varíaveis H e S são invariantes a variações de intensidade de luz e a variável V, que representa a intensidade, varia. Elgammal *et al.* em [EHD00] usam coordenadas de crominância baseadas no espaço RGB normalizado. Neste, dadas as três variáveis de R, G e B, as coordenadas de crominância são dadas por:

$$r = \frac{R}{R+G+B}, \qquad g = \frac{G}{R+G+B}, \qquad b = \frac{B}{R+G+B} \tag{2.3}$$

Uma vez que $r+g+b=1$ bastam duas variáveis para descrever o espaço de cor, temos então o espaço de crominância (r,g). Na figura 2.2 podemos comparar o resultado da deteção utilizando o espaço de cor (R,G,B) e (r,g), comprovando assim que a utilização deste espaço de cor retorna uma boa supressão da sombra da pessoa detetada.

(a) Imagem Original. (b) Resultado da deteção usando o espaço (R,G,B). (c) Resultado da deteção usando o espaço (r,g).

Figura 2.2: Supressão de sombras [MHKS11].

Embora o uso da crominância ajude a suprimir a deteção de sombras, tem o inconveniente de perder informação da intensidade da cor. Se considerarmos uma pessoa a caminhar com uma camisa branca com um plano de fundo cinzento, a pessoa não será detetada uma vez que o branco e o cinzento têm a mesma crominância. Por esta razão, é necessário utilizar sempre uma variável de intensidade da cor. No caso do espaço HSV esta é a variável V, enquanto que no espaço (r,g) é usada uma terceira variável de intensidade $s=R+G+B$, juntamente com r e g. Enquanto estas duas não variam sobre uma sombra, a variável s irá variar com a presença de sombras e zonas mais iluminadas.

A maioria das abordagens para supressão de sombras que se movimentam utilizam este raciocínio de separar as distorções provocadas pela crominância das distorções provocadas pela intensidade da cor.

2.1.2 Deteção baseada no objeto

Como foi discutido na secção 2.1.1 técnicas de subtração do plano de fundo são eficientes, no entanto, apenas em situações em que a câmara é estática. Em muitos cenários de análise de imagem a câmara é móvel: pense-se nos casos de uma câmara montada num robô. Nestes casos, a modelação do fundo não é viável, passando a ser necessário usar uma abordagem orientada ao objeto para eliminar a assunção de que o cenário terá um plano de fundo constante. Segundo Leibe [MHKS11] pode-se abordar o problema de extração de informação de um objeto usando diferentes níveis de detalhe. Do ponte de vista do rastreio este diz-nos que os objetivos principais de deteção são (a) detetar novos objetos, (b) classificar estes objetos num número de categorias de interesse, e (c) continuar a rastreá-los.

De seguida analisam-se os desafios desta abordagem juntamente com as técnicas mais usadas para atingir deteção com base nas características do objeto.

2.1.2.1 Conceito e desafios

A ideia de rastreio por deteção baseia-se em aplicar um detetor para a categoria de objetos pretendidos, a cada imagem de uma sequência de vídeo, e unir o resultado desta deteção para criar trajetórias.

Para atingir este fim é necessário, em primeiro lugar, detetar de forma fidedigna e eficiente a presença de novos objetos de interesse. Uma vez detetado um objeto e iniciado o seu rastreio é preciso que o sistema consiga distinguir se um objeto detetado numa nova imagem é aquele que se está a rastrear ou um novo, para poder associar essa deteção à trajetória do objeto ou iniciar um novo rastreio. A fim de conseguir esta deteção e associação é necessário construir um modelo de aparência, que por sua vez requer segmentação dos planos da imagem. Por fim, para limitar desvios na deteção, a segmentação tem que ser atualizada ao longo do tempo.

2.1.2.2 Abordagens à deteção de objetos

O detetor mais simples é o de *janela deslizante*, no qual uma janela de deteção de tamanho fixo é movimentado sobre toda a imagem, usando um classificador binário em cada localização da janela [MHKS11]. Para que se possam detetar objetos de diferentes tamanhos, a imagem ou a janela de deteção são redimensionadas e o processo é repetido. Usando esta técnica, a deteção de objetos reduz-se a uma simples decisão de classificação binária.

Com o uso de métodos de aprendizagem pode-se reduzir o número de decisões do classificador, o que melhora o seu tempo de execução e reduz o número de falsos positivos.

Uma das abordagens simples à deteção de objetos é a de representar as características de cada janela por um único vetor que codifica o conteúdo da janela em questão. O desafio neste caso passa pela escolha de uma representação suficientemente descritiva para capturar as características da classe do objeto com todas as suas variações e distinguir este do plano de fundo. Em 2005 foi apresentado em [DT05] a representação baseada em histogramas de gradientes orientados (*Histograms of Oriented Gradientes - HOG*). Esta representação divide a janela numa grelha de células, 4×4 ou 8×8, calculando para cada célula um histograma de orientação de gradientes. Em seguida, blocos de células de 2×2 são combinados para normalização do contraste. Para reduzir o efeito do ruído e do processo de quantização, a contribuição de cada pixel é pesada pela magnitude do seu gradiente. Finalmente, todos os blocos na janela de deteção são concatenados num único vetor normalizado. A representação HOG tem várias vantagens: (*a*) o uso de gradientes, ao invés da intensidade do pixel, torna o detetor tolerável a variações de iluminação como sombras; (*b*) a representação por histogramas torna o sistema mais robusto a pequenas variações das regiões da imagem; (*c*) a divisão em grelha adiciona informação localizada e dá mais detalhe à descrição do que o uso de um só histograma; (*d*) a normalização de blocos compensa variações locais de contraste [MHKS11].

Como a representação holística, como HOG, não é capaz de modelar variações locais na estrutura do objeto, como por exemplo diferentes partes do corpo, é necessário um grande número de

exemplos de aprendizagem para que o sistema possa aprender a detetar as alterações de aparência do objeto como um todo.

Uma solução mais flexível é a modelação baseada em partes deformáveis (*Deformable Part-based Model - DPM*) apresentada por Felzenszwalb *et al.* [FGMR10]. DPM representa objetos usando uma mistura de modelos de partes deformáveis a múltiplas escalas. Esta abordagem baseia-se no conceito de estruturas pictóricas, que representam objetos como uma coleção das suas partes organizadas de forma deformável. Cada parte contem propriedades da aparência local do objeto, enquanto que a configuração é caracterizada por ligações entre determinadas partes [FH01]. Assim, o detetor consiste num filtro global, similar ao descritor HOG e num conjunto de filtros das partes, extraídas a resoluções mais altas. O modelo define o valor da hipótese de um objeto como a soma do resultado dos filtros individuais menos o custo da deformação. A aparência das partes do objeto assim como a sua localização são aprendidas automaticamente de dados de treino. Uma vez treinado, este modelo é capaz de detetar o contorno do corpo e os seus membros.

2.1.2.3 Segmentação dos planos da imagem

Como evidenciado anteriormente, um dos desafios do rastreio através de deteção é a necessidade de associar cada nova deteção a um trajetória, descartando falsos positivos. Modelos de aparência, como modelos de cor, geralmente são usados para suportar a associação dos dados e escolher entre vários candidatos. No entanto estes modelos devem ser calculados apenas sobre a região do objeto, enquanto que o detetor de objetos retorna estes juntamente com o seu plano de fundo. Assim, para um rastreio eficiente, é necessário separar o plano de fundo do objeto detetado.

A abordagem mais simples, usada para rastreio de pedestres por Liebe *et al.* [ELSvG08], passa por representar a forma do objeto por uma elipse de tamanho fixo dentro da caixa de deteção. Para detetar o mínimo possível do plano de fundo, a elipse usada por Liebe foca-se na parte superior da pessoa detetada, estendendo-se apenas ligeiramente para a zona das pernas, de forma a cobrir a maior parte possível da pessoa.

Como a elipse usada por Liebe não cobre os membros da pessoa detetada, um método de segmentação mais detalhado é preferível. Uma vez que temos a elipse, que claramente pertence à pessoa e não ao fundo, podem-se usar os pixels do seu interior para estimar a distribuição de cor do objeto, os pixels fora da elipse contêm uma estimativa da distribuição do plano de fundo. Estas duas distribuições são utilizadas como entradas de um sistema de segmentação do tipo *bottom-up* que tentará refinar o contorno do objeto. Neste tipo de abordagem, apesar da inicialização ser dada pela caixa do objeto, a segmentação não requer nenhum conhecimento à priori da forma do objeto, o que a torna aplicável a muitas categorias de objetos articulados diferentes. No entanto, a existência de sombras pode gerar uma estimação errada da distribuição do fundo, devolvendo segmentações incompletas.

Uma abordagem alternativa baseia-se em estimar a segmentação específica à classe do objeto, com base no resultado da deteção. Para que seja possível, é necessário que o detetor de objetos tenha sido treinado com exemplos de segmentação dos planos da figura, o que exige maior poder computacional do que uma simples caixa com o objeto.

2.1.2.4 Rastreio

Numa abordagem de rastreio através de deteção pura é necessária informação do detetor de objetos em cada imagem da sequência de vídeo para que se consiga seguir a trajetória do objeto. Mas, uma vez detetado um objeto a sua aparência não se irá alterar muito rapidamente. Tendo como base esta assunção podem-se usar técnicas de rastreio baseado em regiões para manter o rastreio por pequenos períodos. Esta técnica é útil em casos em que objeto pode ficar tapado, ou ocluído, permitindo o seu rastreio, assim como permite diminuir a necessidade computacional do sistema, reduzindo a quantidade de vezes que o detetor de objetos é ativado. Wu e Nevatia [WN06] propuseram uma abordagem a rastreio através de deteção, baseado na aparência, que usa rastreio com deslocamento da média para unir pequenas lacunas quando não existe deteção.

2.2 Língua Gestual

Muitas abordagens ao reconhecimento de Língua Gestual – na literatura inglesa é usado o termo *Sign Language Recognition (SLR)* – cometem o erro de tratar o problema como puramente de reconhecimento de gestos. A Língua Gestual é tão complexa quanto qualquer outra língua, sendo ainda que, na Língua Gestual, o significado é transmitido através de múltiplos canais em simultâneo. Cooper *et al.* [MHKS11] fazem uma simplificação do problema identificando três partes fundamentais da Língua Gestual:

1. *Características Manuais*, que englobam gestos realizados com as mãos, usando a forma da mão e movimento para transmitir um significado;

2. *Características não Manuais*, tais como expressões faciais ou a postura do corpo, que podem formar parte de um sinal ou modificar o seu significado;

3. *Ortografia Gestual*, descrevendo uma palavra, ao usar as suas letras constituintes de forma gestual, no alfabeto local. Na figura 2.3 está presente o alfabeto gestual usado na Língua Gestual Portuguesa.

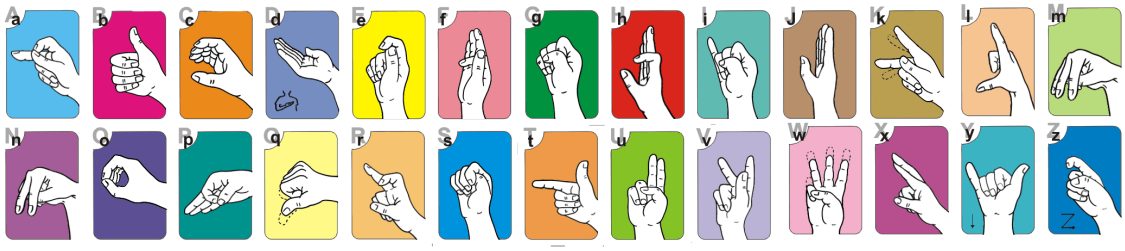


Figura 2.3: Alfabeto gestual usado na Língua Gestual Portuguesa. Adaptado de [dS11].

De agora em diante iremos referir-nos a *gesto*, *sinal* ou *símbolo* como um elemento gestual, composto de características manuais e não manuais, que tem um significado associado. No entanto, como nos focaremos no desenvolvimento de mecanismos para a deteção de ortografia gestual iremos utilizar o termo de gesto estático para nos referir a este. Assim, neste trabalho, os termos "ortografia gestual" e "gesto estático" referem-se à mesma categoria de elementos da Língua Gestual.

2.2.1 Língua Gestual Portuguesa

A Língua Gestual não é universal, sendo uma característica de cada país e cultura [Bal10]. Assim sendo, em Portugal é usada a Língua Gestual Portuguesa. Tendo sido desenvolvida em paralelo com a Língua Portuguesa (LP), LGP não imita a sua contraparte e usa a sua própria sintaxe, tirando partido de características manuais e não manuais, num padrão simultâneo ou sequencial, arranjado no espaço tridimensional. Assim, a sua estrutura é também distinta da usada habitualmente na LP. A sua sintaxe é predominante organizada segundo *sujeito-objeto-verbo* (SOV). Por exemplo a oração "Eu vou para casa" fica, em LGP, como *(Eu) casa ir*. Outra característica, observável no exemplo, é o fato de LGP não usar preposições (e.g.: "a", "para", "em", etc.). Ainda, no caso de o sujeito ser um pronome pessoal, e estiver implícito no contexto, poderá não ser necessário marcá-lo.

Os **verbos** em LGP são sempre realizados no infinitivo. Para marcar o tempo verbal são usados advérbios de tempo ou, na sua ausência, é usado o movimento do corpo, sendo que para a frente indica futuro e para trás indica passado.

A marcação do **género**, na LGP, surge apenas no caso de referência a seres animados, usando, normalmente, recurso aos gestos "homem" e "mulher", como marcas de masculino e feminino, respetivamente. No entanto, normalmente não é marcado o masculino, enquanto que o feminino é marcado por prefixação, isto é, o gesto *mulher* aparece antes do gesto que se pretende fletir em género. Existem ainda alguns casos em que o gesto no feminino e no masculino são diferentes (e.g.: *mãe/pai*).

A marcação de flexão em **número**, como o plural, pode ser efetuada de diferentes formas. Ana Bela Baltazar [Bal10] descreve-as como:

Repetição, quando, para marcar o plural, o gesto é repetido;

Redobro, quando o gesto é realizado por ambas as mãos;

Incorporação, quando se usa um número para especificar quantidades reduzidas (e.g.: "quatro filhos" = *filho* + *quatro*);

Determinativo, usado para descrever quantidades não contáveis (e.g.: "muitos homens" = *homem* + *muito*).

Para realizar uma frase **interrogativa**, é usada a expressão facial que poderá ser combinada com o uso de pronomes interrogativos no final da frase. A frase **exclamativa**, por sua vez, é apoiada pela expressão facial e pela postura do tronco e da cabeça.

A diferença entre sinais é muito grande e cada individuo tem o seu próprio estilo, tal como na escrita. O individuo que pratica LGP terá uma *mão dominante*, como a mão direita para uma pessoa destra, e uma *mão não dominante*, pense-se na mão esquerda no caso anterior. O desempenho entre a mão dominante e a mão não dominante pode variar.

2.2.2 Aquisição e reconhecimento de dados

O primeiro passo num sistema de reconhecimento de Língua Gestual será sempre a aquisição dos dados. A maioria dos primeiros sistemas na área usavam luvas virtuais, como a DataGlove [KE96], e acelerómetros para recolher sinais específicos vindos das mãos. Nestes casos, as medidas como posição no plano (x, y, z), orientação, velocidade, etc, eram retiradas diretamente, sendo que muitas vezes os resultados dos sensores eram suficientemente bons para possibilitarem que fossem diretamente usados como características do sinal [MHKS11]. Embora este tipo de sistemas tenha a vantagem de devolver posições precisas, não permitia uma movimentação natural, restringindo a fluidez natural do movimento, alterando assim o sinal executado. Embora alguns sistemas tenham sido apresentados que reduziam este problema, os custos desta abordagem são geralmente muito elevados, levando ao uso da visão por computador.

Geralmente, no caso de visão por computador, uma sequência de vídeo é capturada usando uma combinação de câmaras. Em 1999 Segen e Kumar [SK99] usaram uma câmara e uma fonte de luz calibrada para calcular profundidade. Em 2004 Feris *et al.* [FTR+04] utilizam, uma série de fontes de luz externas para iluminar o cenário aplicando geometria de vários ângulos de visão para construir uma imagem de profundidade. Numa abordagem diferente, em 1998, Starner *et al.* [SWP98] usam uma câmara frontal em conjunção com uma câmara montada na cabeça do indivíduo, apontada às mãos, para ajudar no reconhecimento de gestos. Imagens de profundidade podem ser conseguidas usando câmaras estereoscópicas, que têm a capacidade de simular a visão binocular humana usando duas ou mais lentes com sensores óticos separados. Este tipo de câmaras foi usada por Munoz-Salinas *et al.* em 2008 [MSMCMCCP01]. Recentemente o sensor Microsoft Kinect veio oferecer uma câmara de profundidade a um preço muito acessível, tornando as imagens de profundidade uma opção viável.

Uma vez adquiridos os dados, estes são descritos através das suas características. Na Língua Gestual, muitas dessas características baseiam-se nas mãos. Em particular, a forma e orientação da mão assim como a trajetória do seu movimento.

2.2.3 Características manuais

O rastreio das mãos não é uma tarefa fácil uma vez que, na Língua Gestual, os movimentos manuais são rápidos produzindo, muitas vezes, segmentos de vídeo desfocados. As mão são objetos deformáveis mudando de pose e posição no espaço. O movimento de uma mão pode ocultar o movimento da outra, assim como pode também ocultar a face do indivíduo [MHKS11].

Nos primeiros trabalhos, a tarefa de segmentação era simplificada com o uso de luvas coloridas. Zhang *et al.* [ZCF+04] utilizou luvas coloridas e a geometria das mãos para detetar a sua

posição e forma. As luvas usadas por Zhang *et al.* estavam codificadas de forma a que os dedos e as palmas das mãos tivessem cores diferentes. Este tipo de luvas diminui a restrição dos movimentos do indivíduo, provocada pelas luvas virtuais, mas não a elimina. Para uma abordagem mais natural, é usado um modelo da cor da pele como no trabalho de Athitsos e Sclaroff [AS03]. Imagawa *et al.* [ILI98] demonstrou que usando a cor da pele obtinha-se uma boa segmentação, conseguindo segmentar as mãos e face do indivíduo com esta técnica e aplicando, em seguida, um filtro de Kalman para o rastreio. Han *et al.* [HAS03] demonstram que com o uso de filtros de Kalman conseguiam tornar esta abordagem robusta a oclusão. Restringindo o plano de fundo a uma cor específica, ou mantendo-o estático, consegue-se simplificar ainda mais esta tarefa. Zieren e Kraiss [ZK04] usaram esta técnica para facilitar a segmentação do plano de fundo.

Imagens de profundidade podem ser usadas para simplificar o problema. Hong *et al.* [HSL07] utilizam um par de câmaras estereoscópicas que, combinadas com outros sinais, permitiram construir modelos da pessoa na imagem. Por sua vez, Fujimura e Liu [FL06], usando a mesma tecnologia, conseguiram segmentar as mãos, embora com a assunção simplista de que as mãos seriam os objetos mais próximos da câmara.

O sensor Kinect veio oferecer aos investigadores desta área um bom meio para rastrear dados, permitindo desempenho em tempo real. Doliotis *et al.* [DSM+11] demonstram que usando este sensor, em vez do seu método anterior baseado na cor da pele, o desempenho do seu sistema aumenta entre 20% a 95%, num conjunto de dados de dez símbolos de números.

2.2.3.1 Forma da mão

Características da forma da mão são muitas vezes ignoradas, seja porque a resolução do vídeo não é suficientemente alta ou porque o poder de processamento é limitado não permitindo processamento em tempo real. Como alternativa, tende-se a aproximar a forma da mão através da extração de características geométricas como o seu centro de gravidade. O uso de luvas virtuais permite descrever a forma da mão em função dos ângulos das articulações e, de uma forma mais genérica, da abertura dos dedos, como foi demonstrado por Vogler e Metexas [VM04].

Com câmaras esteroscópicas, Rezaeis te al. [RVRD08] reconstroem um modelo tridimensional da mão, processando a correspondência de pontos e a estimação de movimento tridimensional, a fim de criar uma trajetória de movimento 3D completa assim como reconhecer a pose das mãos.

Oikonomidis *et al.* [OKA11] usam o sensor Kinect para obter informação da forma da mão em tempo real, otimizando, de seguida, os parâmetros do modelo da mão usando uma variante de otimização por enxame de partículas (*Particle Swarm Optimization – PSO*) a fim de fazer corresponder a pose atual a um modelo. Embora este método consiga transmitir fielmente os parâmetros da mão, requer ainda um passo para extrair um elemento gestual conhecido.

2.2.3.2 Ortografia gestual

A ortografia gestual é uma extensão das características manuais da Língua Gestual, o seu reconhecimento requer uma boa descrição da forma da mão e, em certos Línguas, o seu movimento

[MHKS11].

Com o uso de câmaras estereoscópicas para obter imagens de profundidade, Jennings [Jen99] demonstrou um sistema de rastreio do movimento dos dedos robusto, usando contornos e cores. O sistema usa os contornos retirados de quatro câmaras, imagens esteroscópicas de duas câmaras e cor de uma outra para detetar e rastrear os dedos. Os canais são combinados usando uma estrutura bayesiana.

Pugeault e Bowden [PB11] usaram o Kinect para criar um sistema de reconhecimento de ortografia gestual interativo, orientado à Língua Gestual Americana (*American Sign Language - ASL*). As mãos são segmentadas usando imagens de profundidade e de cor, sendo usados filtros de Gabor para extrair as características da pose e é usada uma técnica de aprendizagem, baseada em várias árvores de decisão, *florestas aleatórias*, para aprender a distinguir entre letras e formas. A ambiguidade entre certas formas é resolvida através de uma interface que permite ao utilizador escolher a letra correta.

2.2.4 Características não manuais

Juntamente com as características manuais, muita informação na Língua Gestual é transmitida através das características não manuais, tais como a expressão facial e pose da cabeça.

O reconhecimento da expressão facial pode ser interpretado diretamente para a Língua Gestual, ou para um sistema de interação humana mais genérico. Algumas expressões, segundo Ekman [Ekm99], são culturalmente independentes como o medo e a tristeza. A maioria da investigação na área de reconhecimento de expressões faciais, não relacionada com o reconhecimento de Língua Gestual, baseia-se nestas expressões, o que faz com que não se traduzam bem para a área em questão, sendo muitas vezes necessárias expressões exageradas para permitir o reconhecimento. Recentemente, investigadores têm trabalhado com conjuntos de dados não tão restritivos. Estas abordagens poderão provavelmente ser adaptadas à área do reconhecimento de Língua Gestual, uma vez que não têm tantas restrições e usam conjuntos de dados mais naturais [MHKS11].

Vogler e Goldstein abordam o problema de rastreio de características faciais no contexto de reconhecimento de Língua Gestual utilizando um modelo deformável da face [VG08]. Estes mostram que ao fazer corresponder pontos ao modelo e categorizando-os como estando dentro ou fora deste, é possível gerir oclusão pelas mãos. Eles propõem que não é necessário rastreio com oclusão completa, mas sim uma "recuperação graciosa". Este conceito sugere que quando a boca do indivíduo está escondida não é necessário saber a sua forma, podendo a informação ser retirada do que acontece antes e depois da oclusão, da mesma forma que um observador humano o faz. Porém esta teoria pode-se revelar muito difícil de comprovar.

2.3 Sumário

O estudo exposto neste capítulo visou abordar os temas relevantes para uma melhor compreensão de como funciona um sistema de deteção e análise, focando-se maioritariamente na deteção e análise de movimento humano.

Começámos por abordar o problema da deteção. Vimos que existem, na literatura, duas grandes abordagens a este problema: a baseada no pixel e a baseada no objeto. Estudámos como cada uma destas técnicas é realizada, analisando os seus maiores desafios e como estes são ultrapassados.

Por fim, centrámo-nos na área de interesse deste projeto: o reconhecimento de Língua Gestual. Começámos por analisar a língua e os elementos que constituem um gesto. Olhámos de seguida para questões linguísticas da Língua Gestual Portuguesa a fim de melhor entender as diferenças entre esta e a Língua Portuguesa. Finalmente, analisámos abordagens a deteção e reconhecimento de cada uma das categorias constituintes de um gesto para aumentar o nosso entendimento das dificuldades na conceção de um sistema de reconhecimento de Língua Gestual.

Capítulo 3

Ferramentas e arquitetura

Neste capítulo pretende-se fazer uma análise das ferramentas usadas neste projeto assim como apresentar a arquitetura do sistema. Começa-se por analisar o sensor Kinect. Este é uma junção de sensores e algoritmos especialmente desenhados para realizar um sistema capaz de detetar e seguir o movimento humano. Apresentam-se algumas aplicações relevantes desenvolvidas com este a fim de dar uma visão das variadas áreas onde o sensor pode ser aplicado. Passa-se então a analisar o seu funcionamento de uma forma mais aprofundada, estudando projetos que usam o sensor. Estudam-se as diferentes formas de captação de imagem assim como a forma como o sensor consegue detetar e seguir movimento humano.

Passamos então a introduzir o pacote de desenvolvimento usado para interagir com o sensor, o KinectSDK assim como a biblioteca *OpenCV* escolhida para o processamento de mecanismos de visão por computador.

Finalmente, no que diz respeito à arquitetura do sistema, apresentam-se diagramas de alto nível para cada uma das aplicações desenvolvidas.

3.1 Microsoft Kinect

Sendo originalmente apresentado como "Projeto Natal" a 1 de Junho de 2009, o sensor Kinect foi lançado a 4 de Novembro de 2010 como um acessório da consola Xbox 360 da Microsoft. Este é o fruto da parceria entre a empresa Israelita PrimeSense e a Microsoft [CLV12].

O sensor Kinect foi criado para servir como uma forma de interação entre o utilizador e a consola Xbox 360, utilizando gestos e comandos de voz. Assim, o sensor é capaz de capturar imagens com 640×480 pixels a $30 fps$. Utilizando informação de profundidade, o sensor é ainda capaz de produzir um modelo do esqueleto da pessoa que está a ser capturada. Com este modelo é possível definir gestos que serão reconhecidos pelo Kinect e usá-los para interagir com o computador.

Em Junho de 2011 a Microsoft lançou um *Software Development Kit* (SDK) para usar o sensor Kinect com o sistema operativo Windows 7, sendo que em Fevereiro de 2012 a versão para Windows do Kinect, *Kinect for Windows*, foi lançada.

3.1.1 Aplicações desenvolvidas com Kinect

O Kinect foi criado para revolucionar a forma como as pessoas interagem com jogos e a sua experiência, podendo interagir de uma forma natural, com o seu corpo [Zha02]. É ainda capaz de receber comandos de voz e consegue identificar utilizadores quando estes se aproximam.

Antes do seu lançamento três demonstrações das capacidades do Kinect foram apresentadas. Estas foram *Ricochet*, *Paint Party* e *Milo and Kate* [CLV12]. Em *Ricochet* um avatar imita todos os movimentos do utilizador e o objetivo deste jogo era acertar em bolas virtuais. *Paint Party* era uma aplicação de pintura, dando a possibilidade ao utilizador de escolher diferentes tipos de pincéis e utilizar gestos para colorir. *Milo and Kate* era a demonstração mais complexa. Criado pelos estúdios Lionhead o jogo funcionava como uma inteligência artificial emocional. O utilizador interagia de uma forma natural com um rapaz virtual de 10 anos, Milo, ou com um cão, Kate. A inteligência artificial do jogo respondia diretamente ao jogador através dos seus gestos, palavras ou ações predefinidas em situações dinâmicas. O sistema "aprendia" com o utilizador, adaptando-se às suas escolhas. Estas aplicações foram apenas usadas para demonstrar as potencialidades do sensor num ambiente de jogo.

Na altura do lançamento, quinze jogos foram apresentados que sairiam para o mercado juntamente com o Kinect, concebidos especialmente para usufruir das novas capacidades de interação oferecidas pelo sensor.

O sucesso do Kinect no mundo dos jogos despertou interesse de investigadores e praticantes de muitas e diferentes áreas como ciências de computadores, engenharia eletrotécnica e robótica. O baixo custo do sensor e as suas capacidades abriam portas para novas formas de interação com diferentes sistemas.

Na secção 2.2 referem-se alguns exemplos de sistemas de reconhecimento de Língua Gestual que usam o sensor Kinect para ultrapassar dificuldades no processamento de imagem e reconhecimento de objetos.

Muitos projetos foram desenvolvidos que usufruem deste sensor, nas mais variadas áreas, de seguida destacam-se alguns.

YScope [YDr12]

A empresa portuguesa YDreams apresentou em 2012 o sistema YScope. Este é um sistema orientado a cirurgiões num ambiente de bloco operatório. O YScope usa o sensor Kinect para permitir que cirurgiões manipulem imagens médicas à distância, mantendo as suas mãos estéreis tanto quanto possível.

Brekel Kinect [Bre12]

Brekel Kinect é uma aplicação que usa o sensor Kinect para permitir a captura de objetos tridimensionais e exportá-los para usar em ambientes 3D. Permite também rastreio do esqueleto para modelação e captura de movimento. É uma aplicação gratuita para uso comercial e privado.

A versão *Pro Body* é especializada na captura de movimentos de indivíduos (*motion capture – MoCap*). Consegue operar em tempo real sem a necessidade de pós-processamento, uma capacidade rara em sistemas MoCap. Suporta ainda a deteção de rotações das mãos, pés e cabeça detetando até 2 indivíduos num cenário.

O software tem ainda a versão *Pro Face* especializada em deteção tridimensional da face, sua posição e rotação.

3Gear Systems [WTK12]

Uma vez que o Kinect é direcionado para trabalhar em capturas de corpo completo, a 3Gear Systems desenvolveu um SDK capaz de detetar gestos complexos produzidos pelas mãos do utilizador. A sua tecnologia traz ao Kinect a possibilidade de reconstruir uma representação precisa dos movimentos dos dedos do utilizador.

De momento o sistema usa dois sensores Kinect para evitar oclusão, sendo estes montados um pouco acima do monitor do computador. Esta montagem, juntamente com a tecnologia, permite construir interfaces interativas que respondem a pequenos gestos confortáveis, ao invés de largos gestos como o abanar dos braços.

SigmaNIL [ArG12]

Similarmente ao 3Gear Systems, SigmaNIL é uma *framework* para visão por computador direcionada a interfaces naturais. É capaz de rastrear com precisão os movimentos dos dedos, a posição da mão, reconhecimento de gestos e rastreio do esqueleto da mão.

O sistema SigmaNIL usa apenas um sensor, suportando qualquer sensor de profundidade, como o Kinect. É capaz de interagir com as bibliotecas de base OpenNI e KinectSDK sendo, ainda, desenhado de forma modular para que se possa adicionar funcionalidades.

O interesse de muitos praticantes de várias áreas levou ao desenvolvimento de comunidades na Internet para discutir projetos que usam a tecnologia do sensor Kinect. Um dos melhores exemplos é a comunidade KinectHacks.net [Kin11]. Apenas uma mês após o lançamento do Kinect esta comunidade já contava com 9 páginas com pequenas descrições de projetos [Zha02]. Este número tem vindo a crescer, tendo, na altura da escrita deste documento, 70 páginas.

O interesse geral e o potencial dos projetos que têm vindo a surgir não foram descuidados pela Microsoft. Em 2012 surgiu o programa *Microsoft Accelerator for Kinect* [Biz12] que é um projeto de apoio a empresas no seu início que usam o Kinect. Tendo recebido centenas de candidaturas de todo o mundo, o programa selecionou onze empresas que receberam apoio da Microsoft para desenvolver os seus produtos. As empresas focam-se em áreas muito variadas desde a interação homem-computador [UI12] à terapia física e cognitiva [Inc12].

3.1.2 Sensor Kinect

O software interno do Kinect foi desenvolvido pela Rare, uma subsidiária da Microsoft Game Studios. Por sua vez, a tecnologia do sensor de profundidade, assim como o seu núcleo de processamento, foram desenvolvidos pela companhia PrimeSense [CLV12]. O aparelho é constituído

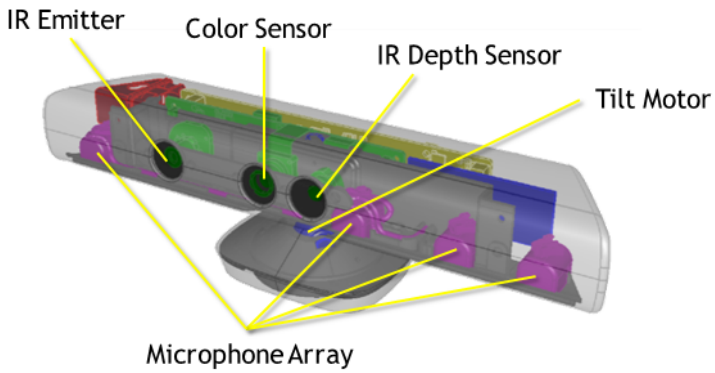


Figura 3.1: Componentes do sensor Kinect [Mic12].

por um sensor de profundidade, uma câmara RGB, um acelerómetro, um motor e uma série de 4 microfones. Na figura 3.1 apresenta-se uma imagem da posição dos constituintes no aparelho.

3.1.2.1 Sensor de profundidade

O sensor de profundidade em si consiste num emissor de infravermelhos e uma câmara de infravermelhos, capaz de detetá-los. O emissor de infravermelhos cria um padrão estruturado de luz infravermelha e a câmara lê a reflexão desses raios. Por sua vez, a câmara interpreta a deformação da projeção e converte essa informação em valores de profundidade, medindo a distância entre o objeto e o sensor. Esta medição baseia-se em triangulação tendo em conta o emissor, a câmara e as posições dos pixels no cenário[CLV12]. Existe uma distância de $7.5cm$ entre o emissor e a câmara de infravermelhos, razão pela qual é necessário calibrá-los para que a medição dos valores de profundidade seja correta.

A câmara de infravermelhos opera a $30fps$, criando imagens de 1200×960 pixels que são decimadas para 640×480 pixels com 11bits, o que resulta numa sensibilidade de $2^{11} = 2048$ níveis. O valor da profundidade é codificado numa escala de cinzentos.Quanto mais escuro for o pixel, mais próximo do sensor está esse ponto no espaço. Pixels pretos indicam que não existe informação de profundidade para esse ponto. Isto ocorre no caso dos pontos estarem muito longe, impossibilitando uma boa medição da sua profundidade, no caso de estarem numa sombra onde não haja pontos do emissor de infravermelhos, no caso de o objeto refletir mal a luz infravermelha (como no caso de espelhos ou cabelo) ou, finalmente, no caso de os pontos estarem muito próximos do sensor, uma vez que o campo de visão do Kinect é limitado devido ao emissor de infravermelhos e à câmara [Zha02].

Uma das alterações introduzidas no sensor na versão para Windows foi a criação do *near mode*, que permite que o Kinect reconheça pessoas e objetos de uma forma mais precisa entre os $40cm$ e os $4m$ [fWT12], sendo que o seu ângulo de visão é de 57^{o} na horizontal e 43^{o} na vertical [Mic12].

3.1.2.2 Câmara RGB, motor, acelerómetro e microfones

A câmara RGB consegue captar imagens de 640×480 pixels, com 8 bits por canal, a $30 fps$. O Kinect pode ainda operar num modo de alta resolução, captando imagens de 1280×960 pixels a $12 fps$ [CLV12].

O motor e o acelerómetro trabalham em conjunto. O motor proporciona um mecanismo para inclinar o aparelho com uma amplitude de $\pm 27^{o}$. Por sua vez, o acelerómetro, configurado a uma amplitude de $2G$, onde G representa a aceleração provocada pela força da gravidade, é usado para determinar a orientação do Kinect [Mic12].

O sistema de microfones é composto por uma série de quatro destes aparelhos. Com os microfones o Kinect é capaz de gravar áudio, determinar a localização da fonte sonora e a direção da onda de áudio [Mic12].

3.1.3 Imagens de profundidade – RGB-D

Pixels numa imagem de profundidade indicam medidas calibradas de profundidade e não uma medida da intensidade da cor. As câmaras com capacidade de analisar a profundidade do cenário apresentam vantagens sobre as câmaras normais. Entre estas vantagens destacam-se a capacidade de operar a baixos níveis de luz, o facto de serem invariáveis à cor e textura assim como o facto de serem capazes de resolver ambiguidades na silhueta de um indivíduo [SFC+11].

A forma mais natural de captura de dados através do Kinect é através de imagens RGB-D. Este tipo de imagens resulta da combinação dos canais de cor vermelho (R), verde (G) e azul (B) com informação de profundidade (D).

A natureza distinta da origem da informação deste tipo de imagens – sendo que as cores têm natureza visual e a profundidade uma natureza geométrica – permite usar esta informação para realizar facilmente tarefas que seriam complexas apenas com imagens RGB, como segmentação de objetos em tempo real e reconhecimento de pose [CLV12].

As imagens de profundidade trouxeram melhoramentos nas áreas de processamento de imagem e visão por computador. Até ao surgimento de sensores a preço acessível como o Kinect, os investigadores estavam limitados no acesso a imagens RGB-D. Da necessidade de alguns investigadores surgiram conjuntos de dados, acessíveis na Internet, que contêm imagens de profundidade, entre os quais se destacam o NYU Depth Dataset [NSF12], o RGB-D Object Dataset [KLF12] e o Cornell-RGBD-Dataset [Sax09].

O uso destes conjuntos de dados permite o uso de imagens de profundidade em investigação mesmo que não se tenha acesso a um sensor Kinect.

3.1.4 Rastreio do esqueleto

Uma das grandes contribuições do Kinect foi a sua capacidade inovadora de reconhecimento e rastreio do movimento humano em tempo real adaptado a diferentes pessoas, de diferentes tamanhos e formas, sem necessidade de calibração. Esta capacidade baseia-se nos avanços feitos por Shotton *et al.* [SFC+11] em rastreio do esqueleto [Zha02].

Rastreio do esqueleto é um processo que representa o corpo humano por um número de articulações representativas de partes do corpo, como a cabeça, o pescoço, os ombros e os braços. Cada articulação é representada pela sua localização no espaço 3D.

Shotton *et al.* desenvolveram um método de prever as posições no espaço tridimensional das articulações do corpo humano, de uma forma rápida e precisa, a partir de uma única imagem de profundidade, sem usar informação temporal: foi desenvolvido um passo intermédio que trata a segmentação das partes do corpo humano como uma tarefa de classificação baseada no pixel. A avaliação em separado de cada pixel evita a necessidade de uma pesquisa combinatória sobre todas as articulações do corpo [SFC+11]. Com este processo, o sistema consegue identificar as diferentes articulações do corpo do indivíduo detetado, criando um esqueleto com as propostas das posições das suas posições no espaço tridimensional.

Na figura 3.2 apresenta-se uma visão geral deste procedimento. No primeiro passo realiza-se a classificação das partes do corpo numa abordagem baseada no pixel, atribuindo a cada pixel uma cor. Cada cor corresponde à probabilidade desse ponto pertencer a uma determinada articulação. É criada uma hipótese da posição de cada articulação no espaço tridimensional ao encontrar o centróide global de cada parte, através de um deslocamento de média. O último passo do processo é o mapeamento das hipóteses das articulações para as do esqueleto, considerando a continuidade temporal e o conhecimento de dados já aprendidos pelo sistema [Zha02].

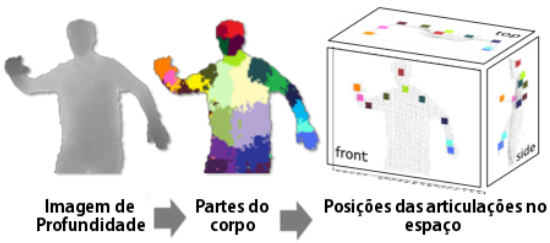


Figura 3.2: Processo de estimação da posição das articulações do corpo humano desenvolvido por Shotton *et al.* Adaptado de [SFC+11].

Para treinar o sistema foram geradas imagens de profundidade realistas de humanos com diferentes formas e tamanhos em diferentes poses. Para melhorar a velocidade de processamento, o classificador pode ainda correr em paralelo na unidade de processamento gráfico (GPU). O algoritmo desenvolvido por Shotton *et al.* consegue correr a $200 fps$ na GPU da Xbox 360. Mais ainda, a abordagem discriminativa aprendida consegue superar casos de auto-oclusão e poses cortadas pela imagem.

3.1.5 Rastreio da posição da cabeça e da expressão facial

A deteção da expressão facial e da posição da cabeça têm sido uma área de investigação ativa na área de visão por computador. A sua realização tem impacto em diversas áreas como interação homem-computador e reconhecimento facial. A maioria das abordagens tende a focar-se em

imagens bidimensionais. A falta de características faciais distintas neste tipo de imagem obriga a explorar técnicas baseadas na aparência e em modelos da forma.

Investigação mais recente foca-se em adaptar modelos deformáveis a digitalizações tridimensionais da face humana [Zha02]. Para o efeito usam-se digitalizadores capazes de captar imagens 3D de alta qualidade usando sistemas de lasers de luz estruturada, produzindo bons resultados, embora a um custo elevado ou sendo necessário muito tempo para produzir uma digitalização.

O sensor Kinect tem a capacidade de juntar vídeo 2D e imagens de profundidade a $30 fps$, a um custo baixo. Porém, a informação de profundidade do Kinect não é extremamente precisa, contendo muito ruído.

Cai *et al.* [CGZZ10] desenvolveram um algoritmo de ajuste de um modelo deformável (*deformable model fitting – DMF*) através de máxima verosimilhança, capaz de lidar com a entrada ruidosa das imagens de profundidade, para ser usado com o Kinect. É usado uma modelo linear deformável da cabeça humana com combinação linear de uma face neutra, uma série de formas básicas com coeficientes estáticos ao longo do tempo, que representam uma pessoa em particular, e uma série de formas básicas com coeficientes, estes dinâmicos ao longo do tempo, que representam as expressões faciais [Zha02].

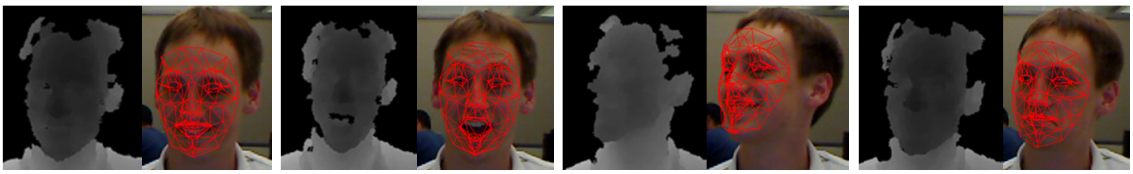


Figura 3.3: À esquerda a imagem de profundidade obtida pelo Kinect e à direita o resultado correspondente do algoritmo desenvolvido por Cai *et al.* [CGZZ10].

Uma vez que o Kinect retira a informação de profundidade através de triangulação, os valores obtidos para esta medida não têm todos a mesma precisão. O erro da medição de profundidade aumenta com o quadrado da distância. Assim, para formular a distância entre o modelo da face e o mapa de profundidade, cada ponto do mapa tem a sua própria matriz de covariância adequada para modelar a sua incerteza. Além disso, as características faciais obtidas através do vídeo bidimensional são rastreadas ao longo das tramas e facilmente integradas na estrutura DMF. Na figura 3.3 podemos observar o resultado do rastreio deste algoritmo.

3.2 Microsoft KinectSDK

O pacote de desenvolvimento KinectSDK foi lançado sob uma licença não comercial em Junho de 2011. O SDK é composto por controladores do sensor compatíveis com o Windows 7 e uma série de bibliotecas específicas que permitem o desenvolvimento de aplicações em C++, C# e Visual Basic.

O pacote de desenvolvimento permite acesso às fontes de dados captados pelos sensores, isto é pelo sensor de profundidade, câmara RGB e 4 microfones do sistema de captura de áudio.

Permite ainda acesso ao resultado do rastreio do esqueleto e oferece capacidades avançadas de processamento de áudio.

Para o desenvolvimento do projeto foi escolhida a linguagem de programação C# por ser uma linguagem orientada a objetos, apresentando elevada versatilidade e capaz de interoperar com outras linguagens através de bibliotecas específicas.

Ainda, na versão para C# o KinectSDK desfruta de um certo nível de abstração das operações necessárias ao controlo do sensor, ao contrário da versão para C++. Esta camada de abstração permite um desenvolvimento mais rápido e evita erros de comunicação com o hardware.

3.3 *OpenCV* e *Emgu CV*

O *OpenCV* (*Open Source Computer Vision Library*) consiste numa vasta biblioteca multi-plataforma de uso livre com mais de 2500 algoritmos orientados a processos de visão por computador e aprendizagem automática (*Machine Learning*). Foi desenvolvido como uma forma de criar uma infraestrutura comum a aplicações de visão por computador e acelerar o uso destes mecanismos em aplicações comerciais [Its13].

O *OpenCV* foi desenvolvido nativamente em C++, tendo interfaces para C, Python e Java. Suporta Windows, Linux, Android e MacOs. A biblioteca foca-se na eficiência computacional dos seus algoritmos, tendo grande relevância em aplicações em tempo real.

Por sua vez, o *Emgu CV* é um *wrapper* (encapsulador) da biblioteca de processamento de imagem *OpenCV*, que possibilita implementar as funcionalidades desta última em linguagens compatíveis com .NET como C# [CV12].

O uso do *Emgu CV* possibilita a implementação de processos de visão por computador otimizados, usufruindo de um certo grau de abstração da sua lógica programática e levando a um melhor e mais rápido desenvolvimento das rotinas necessárias para atingir o nosso objetivo.

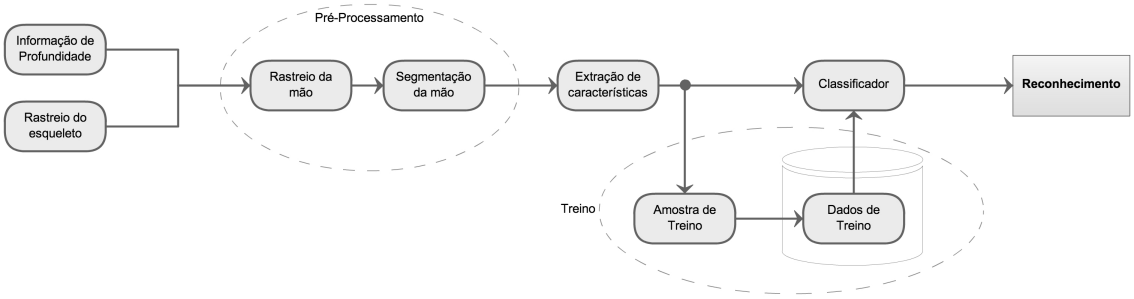
3.4 Arquitetura do sistema

Como mencionado anteriormente, este trabalho pretendia desenvolver métodos simples para reconhecimento de elementos da Língua Gestual Portuguesa, focando-se em dois tipos de elementos: os gestos estáticos e a expressão facial. Para abordar estes problemas, propõe-se uma arquitetura de sistema que será discutida nesta secção.

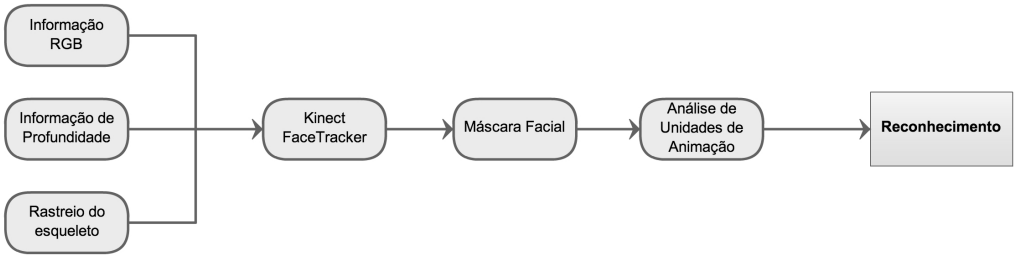
Abordaremos, então a arquitetura de alto nível do sistema e os módulos necessários para que este opere. Uma vez que o sensor Kinect é usado nas duas vertentes do projeto, explica-se aqui também como é feito o processamento da informação captada por este.

A arquitetura proposta é apresentada na figura 3.4, onde são mostrados, separadamente, a arquitetura para o sistema de reconhecimento de gestos estáticos (3.4a) e de reconhecimento da expressão facial (3.4b).

O primeiro diagrama engloba tanto o sistema de reconhecimento de gestos estáticos, em tempo real, assim como o sistema de treino que é necessário para o seu funcionamento. É usada apenas a



(a) Reconhecimento de gestos estáticos.



(b) Reconhecimento de expressão facial.

Figura 3.4: Diagramas de alto nível do sistema.

informação de profundidade e da posição do esqueleto devolvidas pelo sensor Kinect, pois conterão informação suficiente para a deteção do gesto. No entanto, na imagem de profundidade existe informação referente a toda a cena captada pelo sensor. Assim, para um processamento mais eficiente e rápido, é necessário reduzir a quantidade de informação disponível de forma a termos apenas aquela que é mais relevante. A este passo damos o nome de pré-processamento.

Na fase de pré-processamento utiliza-se a informação do rastreio do esqueleto para detetar a posição da mão. Sabendo onde esta se encontra na imagem cria-se uma região de interesse que a englobe e gera-se uma nova imagem contendo apenas a informação desta região. Esta continua a ter informação desnecessária, como informação sobre o plano de fundo. Interessa-nos apenas o primeiro plano onde se encontra a mão do utilizador, portanto passa-se a um processo de segmentação para obter somente a informação da mão.

Concluída a fase de pré-processamento procede-se à extração de características. Aqui a imagem binária resultante do processo de segmentação é analisada e são extraídas as suas características relevantes. Estas são organizadas num vetor formando um padrão descritivo do gesto que o utilizador está a realizar.

O vetor de características resultante pode ser usado de duas formas: pelo sistema de treino e pelo sistema de reconhecimento. No primeiro caso, o vetor torna-se numa amostra da classe (declarada pelo utilizador) do gesto executado e é guardada num ficheiro. Ao conjunto de todos os dados que constituem o ficheiro dá-se o nome de conjunto de treino. Através deste processo consegue-se construir facilmente um conjunto com várias amostras por cada gesto que será utilizado durante o processo de reconhecimento, uma vez que o classificador utiliza toda a informação do conjunto disponível para aprender os padrões de características de cada gesto.

O vetor resultante do processo de extração de características é então passado ao classificador. Este compara o vetor com todas as amostras de treino que tem disponíveis associando-o a uma determinada classe. A resposta do classificador é a classe que este associou ao vetor recebido obtém-se, assim, reconhecimento do gesto efetuado.

No segundo diagrama (figura 3.4b), descreve-se o funcionamento do sistema de reconhecimento de expressões faciais. Neste, é usada a informação RGB, de profundidade e de posição do esqueleto para detetar a posição e orientação da face do utilizador, visto que estas fontes de dados são necessárias ao funcionamento do sistema Kinect *FaceTracker* (que será detalhado na secção 4.1). Neste módulo utiliza-se as capacidades do *Face Tracking SDK* para detetar a cara do utilizador de uma forma simples e eficiente. Esta informação é então utilizada para criar uma máscara que representa o estado e orientação da face do utilizador a partir de pontos e vetores retirados das suas características faciais. A máscara tem a capacidade de organizar determinados pontos (correspondentes a músculos da face) em unidades de animação que traduzem dinamicamente o estado dos músculos. Estas unidades são analisadas perante uma parametrização de expressões faciais que reflete o estado de cada expressão resultando, finalmente, em reconhecimento da expressão facial do utilizador.

Os dois diagramas apresentados refletem a arquitetura interna das aplicações desenvolvidas. Estes são apresentados (e foram implementados) separadamente, no entanto, foram pensados de forma a poderem ser implementados em paralelo, a fim de se conseguir criar um sistema capaz de detetar, em simultâneo, a expressão facial do utilizador e o gesto realizado por este.

A arquitetura do sistema de reconhecimento de gestos estáticos foi também desenhada de forma a poder servir como uma base sobre a qual se pode estender o funcionamento do sistema para reconhecimento de gestos dinâmicos representando, assim, o primeiro nível de um sistema de reconhecimento de elementos gestuais.

3.5 Sumário

Neste capítulo analisámos o sensor Kinect. Este sensor desenvolvido pela Microsoft trouxe ao mercado uma ferramenta a preço acessível capaz de captar imagens, vídeo e imagens de profundidade, tendo ainda a capacidade de efetuar deteção de movimento em tempo real. Começámos por estudar áreas em que este sensor é usado na atualidade, passando seguidamente a analisar o sensor em si e os seus componentes. Finalmente, estudámos como o Kinect capta imagens de profundidade e como realiza rastreio e deteção.

Apresentámos o KinectSDK, o pacote de desenvolvimento criado pela Microsoft para auxiliar na criação de aplicações que utilizam o Kinect. Com a versão para a linguagem C# deste SDK obtém-se uma maior abstração do controle das fontes de dados do sensor, otimizando assim os mecanismos de baixo nível. Para todos os processos de visão por computador, o *Emgu CV* é uma boa biblioteca direcionada para C#, servindo como ligação a *Open CV*. Assim, temos acesso a algoritmos complexos otimizados de visão por computador e *machine learning*.

Finalmente apresentou-se a arquitetura das duas aplicações que foram desenvolvidas. Os diagramas que foram apresentados servem de suporte à discussão que se segue sobre o trabalho desenvolvido.

Capítulo 4

Desenvolvimento

Neste capítulo aborda-se o trabalho desenvolvido para atingir os objetivos propostos para este projeto, abordando as duas vertentes da Língua Gestual Portuguesa estudadas: a expressão facial e ortografia gestual.

Inicia-se este capítulo por expor o trabalho desenvolvido para atingir o reconhecimento da expressão facial do utilizador. Para o efeito, começamos por estudar as ferramentas empregues que suportam o funcionamento da aplicação realizada, culminando na explicação do funcionamento desta.

De seguida passa-se a analisar os procedimentos necessários para atingir o reconhecimento de gestos estáticos. Focamo-nos na ortografia gestual, isto é, o sistema reconhece letras do alfabeto gestual usadas em LGP. Nesta secção, começamos por expor o método de pré-processamento, passando de seguida à explicitação de características utilizadas e à sua extração. De seguida a expõem-se os algoritmos de classificação utilizados, explicando o seu funcionamento. Finalmente, apresentam-se as duas aplicações desenvolvidas para atingir o reconhecimento de gestos estáticos: a aplicação de treino, que serve para recolher amostras que irão servir para que o sistema reconheça os padrões de cada gesto e a aplicação de deteção de gestos em tempo real.

4.1 Reconhecimento da expressão facial

Começamos por discutir o trabalho realizado para o sistema de reconhecimento da expressão facial. Para tal, introduz-se primeiro a ferramenta de desenvolvimento *Face Tracking* e a parametrização da face CANDIDE. O reconhecimento da expressão facial implementado depende fortemente destas duas ferramentas, sendo-lhes então devida alguma atenção nas seguintes secções.

4.1.1 Microsoft Face Tracking SDK

O pacote de desenvolvimento *Face Tracking*, ou *Face Tracking SDK*, desenvolvido pela Microsoft, acrescenta ao KinectSDK para Windows a capacidade de criar aplicações que conseguem detetar e seguir a face, e sua morfologia, em tempo real.

Este SDK analisa as entradas do sensor Kinect, calcula a pose da cabeça do utilizador e a sua expressão facial, disponibilizando esta informação de forma a poder ser utilizada numa aplicação [MSD12].

O sistema usa a informações da câmara RGB e do sensor de profundidade do Kinect como entradas e, assim sendo, a qualidade da sua resposta é afetada pela resolução destas. Para que o rastreio seja o mais preciso possível, são necessárias entradas com a melhor definição. Com esta restrição em mente usou-se o modo de alta resolução do sensor, com imagens a 1280×960 pixels, sofrendo o malefício de reduzir a capacidade de processamento a apenas $12\,fps$. Da mesma forma, faces próximas do sensor, ou seja, faces grandes, retornam melhores resultados que faces pequenas, isto é afastadas do sensor.

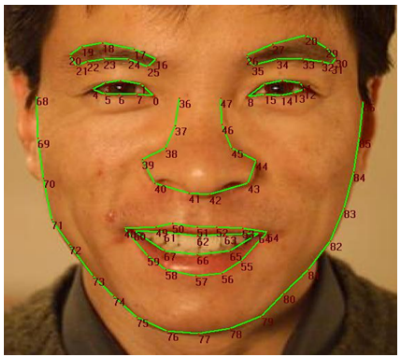


Figura 4.1: Pontos rastreados (13 não são apresentados) pelo *Face Tracking SDK* [MSD12].

O sistema é capaz de rastrear um total de 100 pontos no espaço bidimensional da imagem de vídeo. Os pontos de rastreio são apresentados na figura 4.1, com exceção de 13 pontos que correspondem ao centro do olho esquerdo (87), centro do olho direito (88), centro do nariz (89), interior da sobrancelha esquerda (90-94) e da sobrancelha direita (95-99).

Estes pontos são re-mapeados para o espaço tridimensional resultando num total de 121 pontos, que podem ser observados na figura 4.3, juntamente com a máscara poligonal CANDIDE-3 que será abordada na secção seguinte.

A posição da cabeça do utilizador é traduzida num sistema de coordenadas baseado na regra da mão direita, composta pelos vetores Z, Y e X. O vetor Z tem origem no sensor Kinect e aponta na direção do utilizador, o vetor Y aponta para cima e o vetor X aponta para os lados. A pose da cabeça é traduzida por três ângulos: *pitch*, *roll* e *yaw* (ver figura 4.2). Os ângulos são expressos em graus na gama de -180^{o} a 180^{o}.

Na tabela 4.1 as gamas de valores para cada ângulo são apresentadas tal como é descrito em [MSD12]. São também expostas as gamas de valores aconselháveis para um rastreio bem sucedido com o *Face Tracking SDK*.

4.1.2 CANDIDE-3

CANDIDE é uma máscara parametrizada da face desenvolvida especificamente para a codificação de faces humanas baseada num modelo. Este tipo de máscara usa um número baixo de

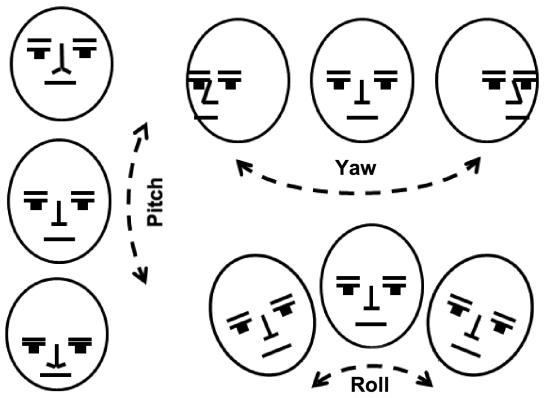


Figura 4.2: Ângulos usados pelo *Face Tracking SDK* para traduzir a pose da cabeça do utilizador [MSD12].

polígonos, geralmente por volta de 100, para ser possível uma reconstrução rápida da face com baixo requisitos de poder computacional.

A máscara CANDIDE é controlada por unidades de animação (em inglês *Animation Units* – AUs) globais e locais, sendo que as globais correspondem a rotações em torno dos eixos X, Y e Z, enquanto que as locais controlam a face de forma a que diferentes expressões possam ser obtidas.

Este modelo foi originalmente desenvolvido por Mikael Rydfakj em 1987 [Ryd87] motivado pelas primeiras tentativas de realizar compressão de imagem através de animação. Esta primeira versão continha 75 vértices e 100 triângulos. Esta variante do modelo é raramente usada.

A iteração CANDIDE-1 surge como uma ligeira modificação do modelo original contendo 79 vértices, 108 superfícies e 11 unidades de animação. Esta foi criada por Mårten Strömberg aquando da implementação do primeiro pacote de software CANDIDE.

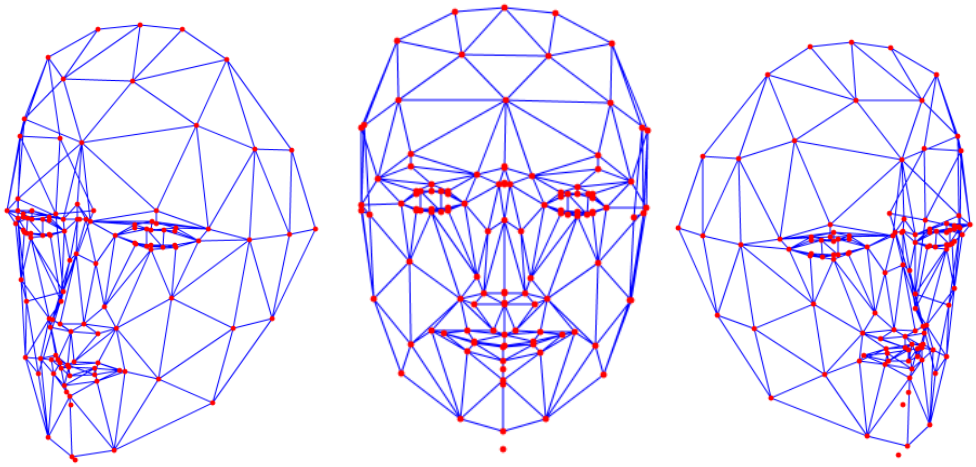
Em 1991 Bill Welsh [Wel91] criou uma nova adaptação do modelo com 160 vértices e 238 triângulos, visando cobrir toda a face frontal e os ombros. Esta versão, conhecida como CANDIDE-2 é distribuída com apenas 6 unidades de animação.

Em 2001 J. Ahlberg [Ahl01] trabalhou sobre o sistema CANDIDE-1 para facilitar a animação de parâmetros faciais da norma MPEG-4 [ISO99] e melhorá-lo com a adição de alguns vértices para aumentar a qualidade deste. Assim surge o modelo CANDIDE-3, contendo 113 vértices, 11 unidades de animação e 12 unidades de forma (*Shape Units – SUs*). Uma unidade de forma define a deformação de uma face padrão para atingir uma face específica, sendo invariante durante o tempo mas específica para um determinado indivíduo. Assim, o parâmetro de forma descreve a forma estática de uma face enquanto que o parâmetro de animação descreve a forma dinâmica desta.

Devido à boa qualidade e flexibilidade desta última evolução do modelo CANDIDE o *Face Tracking SDK* implementa-o como uma forma de conseguir a modelação da face de uma forma fácil e rápida. Na imagem 4.3 pode-se observar os pontos de rastreio (e a máscara poligonal resultante) que perfazem o modelo CANDIDE-3, usando o pacote de desenvolvimento.

Tabela 4.1: Gama de valores de *pitch*, *roll* e *yaw* retornados pelo *Face Tracking SDK*. Adaptado de [MSD12].

Ângulo	Valor
Pitch 0 = neutro	−90 = olhar para baixo, para o chão; +90 = olhar para cima, para o teto; O sistema opera enquanto o ângulo *pitch* da cabeça do utilizador é menor de 20 graus, mas retorna melhores resultados enquanto este for menor que 10 graus.
Roll 0 = neutro	−90 = paralelo com o ombro direito; +90 = paralelo com o ombro esquerdo; O sistema opera enquanto o ângulo *roll* da cabeça do utilizador é menor de 90 graus, mas retorna melhores resultados enquanto este for menor que 45 graus.
Yaw 0 = neutro	−90 = virado em direção ao ombro direito; +90 = virado em direção ao ombro esquerdo; O sistema opera enquanto o ângulo *yaw* da cabeça do utilizador é menor de 45 graus, mas retorna melhores resultados enquanto este for menor que 30 graus.



(a) Vista lateral esquerda. (b) Vista frontal. (c) Vista lateral direita.

Figura 4.3: Pontos de deteção (a vermelho) usados pelo *Face Tracking SDK* projetados no espaço tridimensional. A azul mostra-se a máscara CANDIDE-3.

4.1.3 Unidades de animação

As unidades de animação (AUs) foram anteriormente abordadas por serem uma parte integrante do modelo CANDIDE. Visto que estas unidades é que são efetivamente utilizadas no projeto desenvolvido para modelar e detetar expressões faciais, passamos então a um estudo mais aprofundado das mesmas e da sua utilização com o *Face Tracking SDK*.

O conceito de unidades de animação foi inicialmente descrito pelo investigador sueco Carl-Herman Hjorstjö [Hjo69] em 1969. O seu trabalho foi estendido em 1977 por Paul Ekman e Wallace V. Friesen resultando no atual FACS, *Facial Action Coding System* [EF77]. Com este sistema consegue-se codificar manualmente qualquer expressão facial anatomicamente possível, ao decompo-las nas AUs específicas.

O sistema FACS classifica movimentos faciais humanos através da aparência da face, usando para o efeito unidades de animação. Estas representam contração ou relaxamento de um músculo ou uma combinação de músculos. Assim, são independentes de interpretação, podendo ser usadas para para qualquer tipo de processo de tomada de decisão, incluindo o reconhecimento de emoções básicas.

O modelo CANDIDE-3 tira partido das AUs para modelar facilmente qualquer expressão facial. O *Face Tracking SDK* disponibiliza 6 AUs e 11 SUs, que perfazem um subconjunto do definido no modelo CANDIDE-3. As SUs estimam a forma da cabeça do utilizador, representando assim a posição neutra da sua boca, sobrancelhas, olhos, etc. As unidades de animação, por sua vez, apresentam-se como variações da face neutra e podem ser usadas para manipular pontos num modelo, de forma a que este se comporte da mesma forma que o utilizador.

Na tabela 4.2 apresentam-se as unidades de animação que o *Face Tracking SDK* suporta. Cada AU é normalizada relativamente à face neutra do utilizador e expressa como um valor numérico compreendido entre -1 e 1. Mostra-se também, na tabela, uma breve explicação da relação dos valores de cada unidade de animação com o efeito que esta produz no modelo. O nome descritivo de cada unidade é mantido em inglês para uma ligação mais coerente com a respetiva AU no modelo CANDIDE-3.

Tabela 4.2: Unidades de animação rastreadas pelo *Face Tracking SDK*. Adaptado de [MSD12].

Valor e Nome da AU			Interpretação
AU0 – Upper Lip Raiser (CANDIDE-3 : AU10)	0	=	neutro, a cobrir o lábio;
	1	=	a mostrar os dentes completamente;
	-1	=	lábio puxado para baixo, o máximo possível.
AU1 – Jaw Lowerer (CANDIDE-3 : AU26/27)	0	=	maxilar fechado;
	1	=	maxilar completamente aberto;
	-1	=	maxilar fechado, o mesmo que 0.
AU2 – Lip Stretcher (CANDIDE-3 : AU20)	0	=	posição neutra;
	1	=	lábios completamente esticados;
	-0.5	=	lábios arredondados (a fazer beicinho);
	-1	=	lábios completamente arredondados (posição de beijo).
AU3 – Brow Lowerer (CANDIDE-3 : AU4)	0	=	posição neutra;
	1	=	sobrancelhas totalmente rebaixadas (ao limite dos olhos);
	-1	=	sobrancelhas completamente levantadas.
AU4 – Lip Corner Depressor (CANDIDE-3 : AU13/15)	0	=	posição neutra;
	1	=	lábios muito franzidos (muito triste);
	-1	=	sorriso muito feliz.
AU5 – Outer Brow Raiser (CANDIDE-3 : AU2)	0	=	posição neutra;
	1	=	exterior das sobrancelhas levantado, em expressão de completa surpresa;
	-1	=	exterior das sobrancelhas muito rebaixadas, como numa cara muito triste.

4.1.4 Expressão facial na LGP e sua deteção

A deteção automática de expressões faciais naturais não é uma tarefa fácil. A prova disso é o grande volume de investigação na área e a escassez de soluções robustas que consigam obter bons resultados em tempo real. No entanto, na prática da Língua Gestual a expressão facial opera como um modificador de sentido, alterando o sentido do gesto que se está a fazer para dar origem a uma palavra diferente. Assim, o número de expressões faciais que nos interessa detetar no âmbito deste projeto é limitado. Por outro lado, ao contrário de uma expressão facial natural, que é rápida, a expressão facial usada na Língua Gestual, visto que é marcada durante a execução de um gesto, é mais demorada no tempo, o que vem a facilitar a tarefa.

Ana Bela Baltazar [Bal10] apresenta uma listagem de expressões faciais utilizadas na LGP como um apoio ao seu Dicionário de Língua Gestual Portuguesa. Na tabela 4.3 apresentam-se estas mesmas expressões.

Tabela 4.3: Expressões faciais utilizadas na Língua Gestual Portuguesa. Adaptado de [Bal10].

Bochecha cheia	Bochecha vazia
Boca cerrada	Expressão facial alegre e aberta
Expressão facial triste	Olhar cerrado/expressão facial carregada
Olhar cerrado/expressão calma	Olhar aberto/expressão exclamativa, de admiração ou surpresa
Olhar aberto/expressão de certeza	Expressão interrogativa ou de dúvida
Língua entre os dentes (com som)	

Assim, propusemo-nos provar que com o sensor Kinect e o *Face Tracking SDK* é possível detetar em tempo real expressões faciais simples, como a maioria das utilizadas na LGP. Para o efeito, utiliza-se o sistema inicializando uma instância do *Face Tracking SDK*, ao qual vamos dar o nome de *FaceTracker*.

O sistema foi desenvolvido de forma a que, uma vez inicializado corretamente, o *FaceTracker* deteta a face do utilizador mais próximo do sensor. Tendo sido detetada pelo sistema a face, acedem-se aos pontos rastreados projetados no espaço tridimensional e usam-se estes para criar a máscara poligonal CANDIDE.

Na figura 4.4 podemos observar a interface desenvolvida. Esta é bastante simples, contendo basicamente três áreas relevantes:

1. A região da imagem RGB onde, uma vez detetada a face do utilizador, será representada a máscara CANDIDE;

2. A região à direita contém a imagem de profundidade produzida pelo sensor Kinect, os utilizadores reconhecidos pelo sensor são pintados a cor vermelha (ver secção 4.2.1 para uma maior especificação sobre esta operação);

3. Por baixo da região da imagem de profundidade temos a zona de classificação, onde surge a expressão facial detetada.

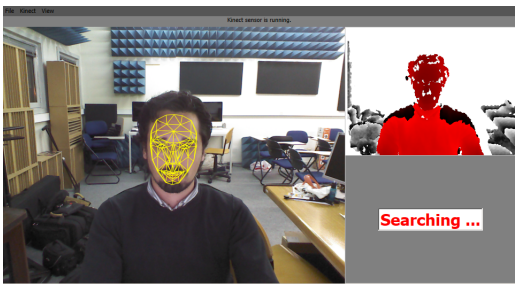


Figura 4.4: Interface desenvolvida para deteção da expressão facial.

Para proceder à deteção de expressões faciais, em primeiro lugar, e de forma a provar o funcionamento do sistema, foram escolhidas duas expressões das representadas na tabela 4.3. As expressões escolhidas foram a expressão facial "alegre e aberta" e a expressão com "olhar aberto/expressão exclamativa, de admiração ou surpresa".

Num primeiro estágio levantou-se a codificação no sistema FACS que caracteriza cada expressão e mapearam-se estes códigos aos mais aproximados disponíveis com o *Face Tracking SDK*, em versão C#. O resultado deste processo pode ser observado na tabela 4.4. A descrição das unidades de animação é mantida, na tabela, com os nomes originais, em inglês, a fim de melhorar a correspondência entre os sistemas. As linhas a branco da tabela traduzem a falta de uma referência direta de unidades de animação entre os sistemas. A partir deste momento, quando nos referirmos a unidades de animação pelo seu índice, estas dizem respeito às unidades de animação do sistema *Face Tracking SDK* (ver tabela 4.2).

Tabela 4.4: Relação entre as unidades de animação do sistema FACS com as disponíveis no *Face Tracking SDK*, para cada expressão facial implementada.

FACS		*Face Tracking SDK*	
AU	**Descrição**	**AU**	**Descrição**
Expressão facial alegre e aberta			
6	*Cheek Raiser*	0	*Upper Lip Raiser*
12	*Lip Corner Puller*	4	*Lip Corner Depressor*
Olhar aberto/expressão exclamativa, de admiração ou surpresa			
1	*Inner Brow Raiser*		
2	*Outter Brow Raiser*	5	*Outter Brow Raiser*
5B	*(Slight) Upper Lid Raiser*		
26	*Jaw Drop*	1	*Jaw Lower*

Tendo as unidades de animação necessárias para parametrizar cada expressão facial passou-se a uma fase experimental onde se observou o comportamento das unidades para cada expressão, em tempo real. Assim, fizeram-se para cada expressão facial algumas assunções, tendo em conta a codificação FACS de cada expressão:

- Expressão exclamativa/surpresa:
  1. Os olhos estão completamente abertos (em expressão de surpresa);
  2. O maxilar está caído isto é, a boca encontra-se aberta.
- Expressão alegre e aberta:
  1. Os lábios encontram-se esticados e numa posição de sorriso;
  2. Os dentes encontram-se parcialmente à mostra.

É relevante voltar a mencionar que, ao contrário de expressões faciais naturais, as usadas na língua gestual não transmitem nenhuma emoção específica, servindo apenas como modificador do sentido do gesto e sendo assim, as assunções acima fazem sentido.

O passo final deste processo foi o de criar limites nos valores das unidades de animação de forma a que assim que fossem ultrapassados o sistema reconhecesse uma determinada expressão facial, tendo em conta os resultados dos valores das unidades de animação na expressão do utilizador. Estes limiares foram procurados experimentalmente.

$$\text{Expressão facial alegre} \begin{cases} AU0 > 0.8 \\ AU4 < -0.2 \end{cases} \tag{4.1}$$

$$\text{Expressão exclamativa/surpresa} \begin{cases} AU1 > 0.5 \\ AU5 > 0.1 \end{cases} \tag{4.2}$$

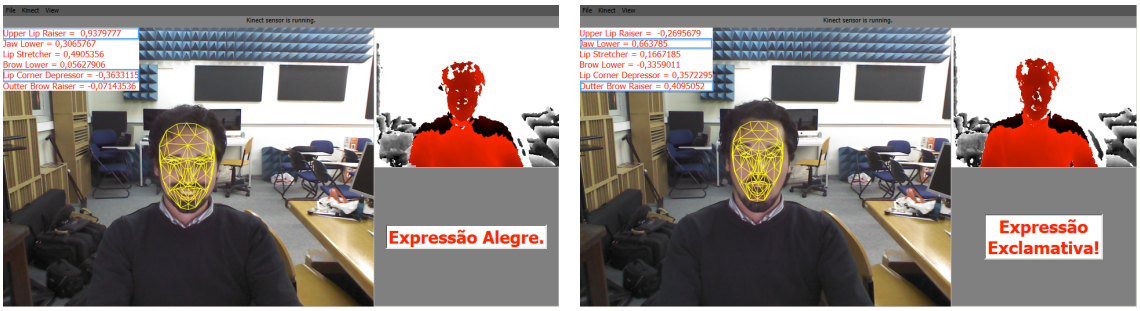
As equações 4.1 e 4.2 mostram os limiares utilizados para a expressão facial alegre e exclamativa, respetivamente. Para que cada uma das expressões seja detetada é necessário que as duas condições se verifiquem simultaneamente.

A figura 4.5 mostra a deteção das expressões faciais implementadas. Apresentam-se também os valores das 6 unidades de animação no momento da captura.

4.2 Deteção e reconhecimento de gestos estáticos

O elemento mais importante numa operação de reconhecimento de Língua Gestual é efetivamente o gesto manual em si. Neste trabalho focamo-nos apenas no gesto estático, isto é, na ortografia gestual (ver figura 2.3).

O gesto estático é isolado no tempo e a sequência de gestos não altera o significado do gesto isolado, o que facilita o seu reconhecimento. Neste trabalho pretende-se apenas provar que é possível desenvolver sistemas capazes de reconhecer todas as vertentes da LGP de uma forma simples



(a) Expressão facial alegre. (b) Expressão facial exclamativa/surpresa.

Figura 4.5: Deteção da expressão facial, com o sistema desenvolvido.

e a baixo custo, utilizando visão por computador, através de um sensor Kinect, sem utilizar nenhum tipo de marcador que dificulte ou altere a interação natural do utilizador. Com este objetivo em mente, desenvolveu-se uma aplicação capaz de reconhecer ortografia gestual em tempo real.

Passamos, nas seguintes secções, a reportar o funcionamento do sistema assim como o trabalho necessário para realizar o nosso objetivo.

4.2.1 Pré-processamento

O sensor Kinect, como referido anteriormente, comporta dois métodos de captação de imagem: um sensor RGB, capaz de captar imagens a cor (*Color Stream*), e um sistema de infravermelhos que é capaz de gerar informação de profundidade (*Depth Stream*). Utilizando a informação de profundidade, o Kinect é ainda capaz de detetar e seguir as articulações do utilizador, dando ao sistema a capacidade de rastrear o seu esqueleto. A esta fonte de dados dá-se o nome de *Skeleton Stream* e contem a localização de todas as articulações do utilizador, no espaço da imagem, assim como a sua distância ao sensor.

No âmbito da deteção de ortografia gestual, a utilização da informação de cor apresenta problemas como a complexidade do processo de segmentação do fundo e a existência de sombras. Assim, a informação de profundidade tem maiores vantagens, facilitando a segmentação dos planos da imagem e eliminando quase por completo os problemas da variação das condições de luz. Para o sistema desenvolvido usou-se apenas o *Depth Stream* e o *Skeleton Stream*.

A maior resolução com que o sensor Kinect é capaz de produzir informação é de 640×480 pixels. A relativamente baixa resolução da captação é contrabalançada com a capacidade de processar imagens a $30 fps$, permitindo uma interação fluída com atrasos pequenos e quase indistinguíveis pelo utilizador.

A partir do momento que se ativam o *Depth Stream* e o *Skeleton Stream*, é necessária uma forma de sincronizar a informação destas duas fontes. Tal é conseguido através da capacidade do sistema de ativar eventos sempre que a informação de um *Stream* esteja disponível. Assim, se a qualquer momento a informação das duas fontes está disponível em simultâneo, estas estão sincronizadas. Caso apenas uma das fontes contenha informação válida esta é descartada. Esta última hipótese ocorre poucas vezes e não afeta a naturalidade da interação.

Como em qualquer sensor de imagem, o campo de visão da câmara de profundidade tem uma forma piramidal. Assim, objetos mais distantes do sensor possuem maior alcance lateral do que aqueles mais próximos. Isto quer dizer que dimensões de pixels de altura e largura não correspondem diretamente a uma posição física no campo de visão da câmara. No entanto, o valor de profundidade de cada pixel traduz diretamente uma distância ao sensor no seu campo de visão. Cada pixel de uma imagem de profundidade é representado por 16 bits, onde os 3 primeiros bits correspondem ao índice do utilizador.

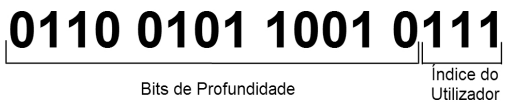


Figura 4.6: Bits de um pixel de uma imagem de profundidade. Adaptado de [WA12].

O KinectSDK tem a capacidade de analisar a informação de profundidade e detetar formas humanas, em um máximo de 6 utilizadores de cada vez, atribuindo uma identificação numérica a cada um dos utilizadores na imagem. Este número é armazenado nos primeiros três bits de cada pixel da imagem de profundidade. Os restantes 13 bits do pixel contêm a informação de profundidade (ver imagem 4.6). O índice do utilizador será um algarismo entre 1 e 6. Se o pixel não fizer parte de uma forma humana (ou utilizador) na imagem, o valor do índice é 0. Para o valor do índice ser preenchido é necessário ativar o *Depth Stream* e o *Skeleton Stream*, caso contrário o SDK não irá detetar utilizadores [WA12]. O *Skeleton Stream*, portanto, consegue detetar até 6 utilizadores mas, no entanto, só consegue rastrear o movimento de um máximo de 2 destes de cada vez.

Com a informação contida no *Depth Stream* conseguimos criar uma imagem na qual a intensidade de cada pixel traduz a distância do elemento ao sensor. Uma vez que cada pixel contem informação que diz se este faz parte, ou não, de um utilizador, usou-se esta capacidade para colorir o utilizador na imagem de profundidade. Este processo ajuda o utilizador a perceber, de uma forma rápida e eficiente, se o sistema o está a detetar efetivamente.

A versão para Windows do sensor Kinect permite dois modos de deteção: normal e o *near mode* (ver secção 3.1.2.1). Uma vez que nesta aplicação nos interessa a posição dos braços e cabeça, e não do restante corpo, assim como boa resolução de imagem, é preferível um modo de operação que consiga detetar o utilizador quando este está mais próximo do sensor. Assim, optou-se pelo segundo modo de operação. Permitindo que o utilizador se encontre mais próximo do sensor, aumenta-se o número de pixels que o descrevem na imagem, aumentando também a definição e qualidade da captação. A partir da versão 1.5 do KinectSDK o *Skeleton Stream* tem também dois modos de rastreio:

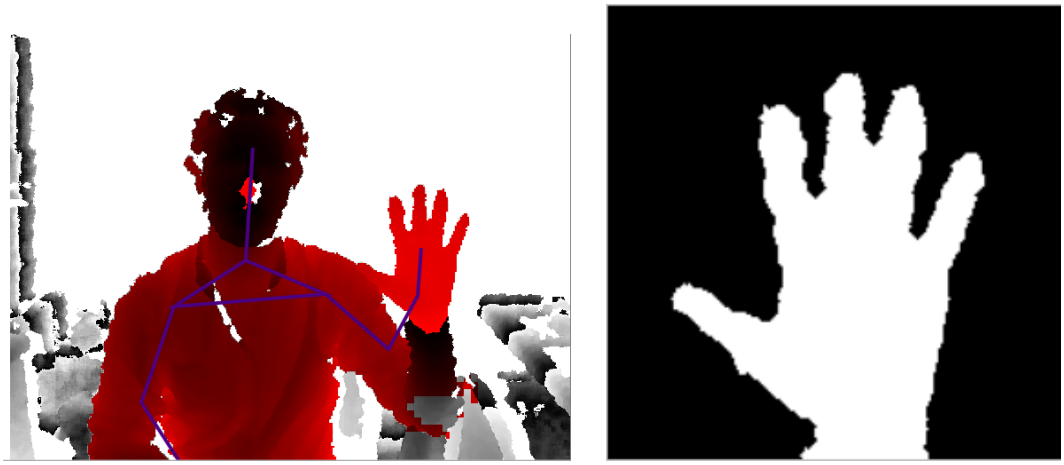
- Modo sentado – rastreia as 10 articulações superiores do utilizador (ombros, cotovelos, pulsos, braços e cabeça);
- Modo por omissão – rastreia 20 articulações do esqueleto, as 10 superiores e 10 inferiores (espinha, quadris, joelhos, tornozelos e pés).

O processo de segmentar uma pessoa sentada é mais complexo que o de o fazer para uma pessoa que está em pé, o que torna o modo sentado mais pesado que o modo por omissão. Não obstante, este é o melhor modo para reconhecer um esqueleto quando o sensor Kinect opera no *near mode*. A acrescentar a isto, o modo por omissão não devolve bons resultados se o utilizador se encontrar sentado, enquanto que o modo sentado opera bem se o utilizador se encontrar em pé. Logo, o modo sentado funciona bem nas duas possibilidades. Assim, e considerando estes aspetos, utilizou-se o modo sentado para a deteção do esqueleto.

Para cada frame produzida pelo *Skeleton Stream* é retirada a informação das 10 articulações rastreadas dos utilizadores e é adicionada, à imagem de profundidade, a estrutura do esqueleto. Esta estrutura ajuda o utilizador a reconhecer erros que estejam a ocorrer durante a sua deteção.

Finalmente, como o sistema deve seguir apenas os movimentos de um utilizador, optou-se por escolher aquele que se encontrasse mais próximo do sensor. Assim, a informação do *Skeleton Stream* referente a outras pessoas presentes no campo de visão do sensor é descartada.

Na imagem 4.7a apresenta-se o resultado do pré-processamento descrito nesta secção.



(a) Resultado do pré-processamento do *Depth Stream* e *Skeleton Stream* em *near mode*.

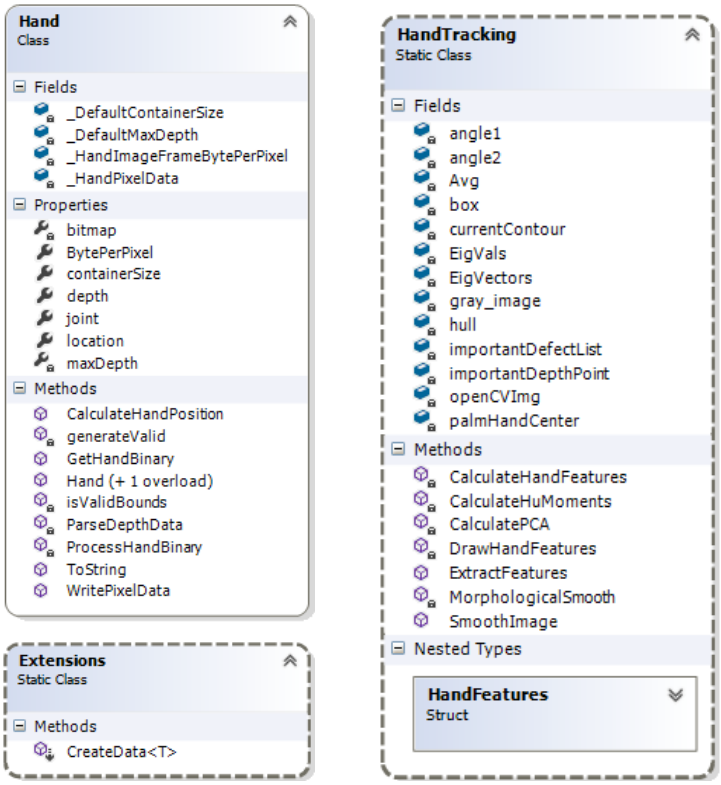
(b) Resultado da segmentação da mão no instante da figura 4.7a.

Figura 4.7: Captura da cena através do *Depth Stream*, pré-processamento e segmentação da mão.

4.2.1.1 Deteção da mão

A ortografia gestual é praticada explicitamente com as mãos. Interessa-nos portanto saber a forma da mão isoladamente. Para tal, é preciso detetar a região da imagem onde esta se encontra.

Utilizando o *Skeleton Stream*, uma vez detetado o utilizador, a posição das suas mãos são dadas pelas suas articulações respetivas. No entanto, esta articulação é um ponto, geralmente centrado na mão. Para detetar a região de interesse da mão e processar toda a informação desta foi criada uma classe *Hand*. A imagem 4.8a representa a estrutura desta classe. A classe *HandTracking* (figura 4.8b), por sua vez, foi criada para o processo de extração de características e será abordada na secção 4.2.2.



(a) Classe *Hand*. (b) Classe *HandTracking*.

Figura 4.8: Diagramas da classes implementadas que operam especificamente com a informação da mão.

Ao inicializar corretamente uma instância desta classe é necessário dar-lhe o comprimento máximo do lado do quadrado que conterá a mão. Através de experimentação encontrou-se um valor aceitável de 250 pixels. Uma vez instanciada a classe e passando-lhe a localização da imagem do pixel onde se encontra a articulação de uma mão, é calculada a área onde toda a mão se encontra. O valor com que a classe foi inicializada servirá de semente ao tamanho inicial do quadrado da região de interesse que englobará a mão.

A classe *Hand* é também capaz de remover, da imagem de profundidade, a informação referente à região de interesse, criando uma nova imagem de profundidade que contem apenas a informação da mão. Essa imagem terá, no máximo, o tamanho do quadrado com que a classe foi inicializada (ou seja, no nosso caso, um tamanho máximo de 250×250 pixels).

O tamanho da região de interesse é diretamente proporcional à distância da mão ao sensor Kinect, uma vez que quanto mais longe do sensor se encontrar o utilizador, menor o número de pixels na imagem que descrevem a mão. Assim, a imagem produzida pela classe *Hand* deverá conter sempre apenas a mão do utilizador, sendo capaz de detetar qualquer uma das mãos (esquerda ou direita). A imagem de menores dimensões ajudará o processamento da informação da mão ao reduzir o número de pixels que é necessário processar, e consequentemente, o número de operações a realizar.

4.2.1.2 Segmentação

Uma vez que a classe *Hand* consegue calcular a zona onde se encontra a mão do utilizador, precisamos de segmentar os planos da imagem. A tarefa está simplificada, primeiro devido à natureza da imagem de profundidade, segundo porque teremos no primeiro plano a mão e a restante informação pode ser descartada.

A ortografia gestual é efetuada apenas pela mão numa posição estática, geralmente posicionada algures à frente do utilizador. Também, o utilizador que será rastreado pelo sistema é aquele que se encontra mais próximo do sensor. Portanto, não deverá existir nenhum objeto entre o sensor e a mão do utilizador, porque caso contrário estaríamos numa situação de oclusão da mão e não há forma de saber o gesto que a mão está a desenhar.

Assim, a segmentação dos planos pode ser alcançada com um simples processo de limiarização de profundidades. Isto é, criam-se dois limiares baseados na profundidade do ponto da articulação da mão. Todos os pixels que se encontrem no interior desses dois limiares fazem parte da mão e por isso tomam o valor binário 1, os restantes pixels tomam o valor 0.

Como o ponto da articulação da mão se encontra à superfície desta, temos que considerar que é mais relevante a informação entre o sensor e a mão do que informação para além desta, isto porque os dedos estarão, no pior dos casos, a apontar na direção do sensor, e nunca para trás. A equação 4.3 mostra os cálculos para os limiares de profundidade superiores e inferiores, onde P_i representa o pixel da imagem a ser avaliado, D corresponde à distância do ponto da articulação da mão ao sensor, P é a profundidade de P_i e N_p corresponde a um valor de percentagem a considerar. Para o sistema desenvolvido achou-se, experimentalmente, que o valor $N_p = 0.15$ retornava bons resultados.

$$P_i = \begin{cases} 1 & \text{se } D(1-N_p) < P < D\left(1+\frac{N_p}{2}\right) \\ 0 & \text{restantes casos} \end{cases} \tag{4.3}$$

O resultado da segmentação tem muitos defeitos devido à baixa resolução da imagem original, resultando em contornos com ruído. Para diminuir estas imperfeições o resultado da segmentação é submetido a uma suavização morfológica. Para o efeito, aplica-se uma dilatação morfológica seguida de uma erosão. A este tipo de suavização morfológica dá-se o nome de operação de fecho e suaviza o contorno de um objeto binário ao eliminar pequenas pontes e fechar pequenos buracos [Dav96].

O resultado após segmentação e suavização pode ser observado na imagem 4.7 onde se apresenta a imagem após o pré-processamento e o resultado da segmentação.

4.2.2 Características da mão

A capacidade de se reconhecer objetos numa imagem depende muito da quantidade de informação que se consegue extrair dessa imagem. A extração de características é portanto uma tarefa fundamental para qualquer processo de reconhecimento e depende fortemente do tipo de objeto em questão. No nosso caso focámo-nos em características relativas à mão humana.

A partir do momento que temos a informação da mão captada e segmentada, em tempo real, iniciamos o processo de extração de características. Para auxiliar neste processo foi usada a ferramenta *Emgu CV*. A classe *HandTracking* (ver figura 4.8b) interage diretamente com o encapsulador *Emgu CV*, usando as suas funcionalidades, para levantar as características necessárias para a análise da mão segmentada.

4.2.2.1 Contornos

A deteção do contorno de um determinado objeto consiste no processo que avalia quais são os pontos da imagem que fazem parte da borda de uma região preenchida por esta. Como a mão foi previamente segmentada, encontrar o seu contorno é uma tarefa relativamente fácil. Por exemplo, tendo a imagem segmentada de uma forma binária, pode-se considerar que o primeiro pixel com valor 1 encontrado num varrimento lateral faz parte do contorno do objeto.

Porém, a informação do contorno do objeto é grosseira. Confere ao sistema uma boa descrição da morfologia da mão mas não permite inferir muito sobre a sua pose. É pois necessário complementar esta informação.

Área convexa

Um método útil de aumentar a compreensão da forma, ou contorno, de um objeto passa por analisar o envelope convexo do contorno e os seus defeitos de convexidade [HT+85].

Matematicamente, o envelope convexo de um conjunto finito de pontos P, no espaço euclidiano, é composto pelo menor conjunto convexo que contem P. Visualmente, se pegarmos num elástico e o esticarmos de forma a envolver todos o conjunto P, a forma que toma o elástico, quando estabiliza ao tocar nos pontos do exterior de P, é o seu envelope convexo.

Os defeitos de convexidade, por sua vez, definem-se como o conjunto de pontos que fazem parte do envelope convexo mas que não fazem parte do conjunto P. A imagem 4.9 ilustra o conceito de defeitos de convexidade usando a imagem de uma mão. Na imagem a linha a vermelho representa o envelope convexo, enquanto que as regiões A a G representam os defeitos de convexidade.

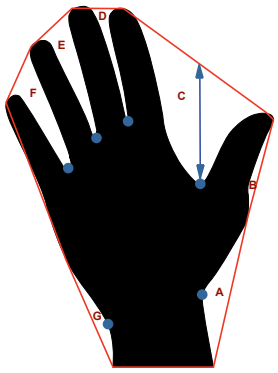


Figura 4.9: Envelope convexo (linha vermelha) e defeitos de convexidade.

Análise de defeitos de convexidade

A análise dos defeitos de convexidade, juntamente com a sua área convexa, dá-nos a capacidade não só de descrever a mão em si mas também o seu estado.

Na imagem 4.9 além dos defeitos de convexidade (*A* a *G*), são representados a azul os pontos do defeito que se encontram mais distantes do envelope convexo. A estes damos o nome de pontos de profundidade que, no caso da mão, geralmente coincidem com os vales entre cada dedo. Podemos também considerar os pontos de início e de fim de cada segmento de linha que delimita um defeito de convexidade, que deverão corresponder a pontas de dedos.

No levantamento de defeitos de convexidade de um objeto maleável como a mão surgem alguns defeitos que não trazem informação relevante, podendo até piorar a compreensão do contorno. O defeito *B* na imagem é um exemplo disso. Se considerarmos que entre cada dois pontos de profundidade encontraremos um dedo da mão, defeitos como *B* dificultam tal assunção. Devemos, portanto, encontrar algumas métricas que nos permitem descartar defeitos de convexidade de forma a termos apenas informação relevante.

Consideramos portanto o maior comprimento entre o ponto de profundidade de um determinado defeito e o seu ponto de fim, o qual chamamos *L*. Se este comprimento, em outro defeito de convexidade, for menor que uma determinada percentagem de *L* esse defeito é descartado. Assim, definimos um valor mínimo do comprimento dos defeitos a considerar. Acertando corretamente o valor da percentagem conseguimos obter os defeitos que correspondem apenas a dedos esticados de uma mão. Sabemos então quantos dedos a mão apresenta levantados e, como o comprimento entre o ponto de profundidade de um defeito e o seu ponto de fim se aproxima muito ao tamanho do dedo, temos também o comprimento de cada dedo.

Centro da palma da mão

A localização, comprimento e estado dos dedos de uma mão são características importantes para compreender a forma de uma mão. No entanto, a localização do centro da palma da mão é uma característica importante que confere informação muito relevante.

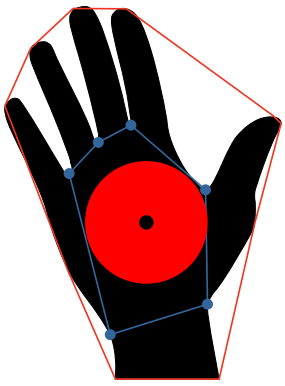


Figura 4.10: Ilustração do maior círculo que cabe na área convexa formada pelos defeitos relevantes.

O centro da palma da mão pode ser aproximado pelo centro do círculo de maior raio que caiba na palma da mão. Portanto, primeiro temos que separar a palma da mão dos dedos. Uma forma de o fazer facilmente é ao considerar os pontos de profundidade dos defeitos de convexidade relevantes. Estes defeitos formam uma área convexa como se pode ver na figura 4.10. Uma vez tendo esta área, basta-nos procurar o centro do maior circulo que caiba nessa e teremos uma aproximação do centro da palma da mão.

A figura 4.11 ilustra o resultado da análise dos defeitos de convexidade, implementada na aplicação desenvolvida. Nesta, a ponta do maior dedo detetado surge a laranja e o centro da palma da mão a vermelho. O objetivo desta medição resulta da necessidade de normalizar o tamanho dos dedos da mão de forma a esta característica ser adaptável a qualquer utilizador. Assim, usa-se o tamanho do maior dedo apresentado para normalizar os restantes comprimentos.

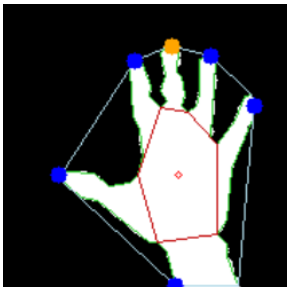


Figura 4.11: Características da mão geradas em tempo real pela aplicação desenvolvida.

4.2.2.2 Momentos

Uma das formas mais simples de comparar dois contornos passa por calcular os momentos da imagem. Estes apresentam-se como uma média pesada da intensidade de todos os pixels. De uma forma lata, o momento é uma característica grosseira do contorno calculada através da integração de todos os pixels deste e multiplicada por uma função escolhida de maneira a conferir um sentido específico [BK08].

Os momentos podem ainda ser normalizados tornando-os invariantes relativamente a rotação, translação e alterações escala, o que torna este tipo de descritor interessante para aplicações de reconhecimento de padrões [GW02].

Quando estamos a lidar com imagens binárias, a integração de uma imagem $I(x,y)$ pode ser compreendida como o somatório de todos os pixels da imagem, que neste caso só poderão tomar o valor 0 ou 1. Assim, o momento (p,q) pode ser definido como:

$$m_{p,q} = \sum_{i=1}^{n} I(x,y)x^p y^q \tag{4.4}$$

Na equação 4.4 p e q representam a ordem da potência à qual as correspondentes componentes x e y serão elevadas. O somatório opera sobre todos os pixels da imagem (n). Segue da análise de 4.4 que o momento $(0,0)$, traduz portanto a área, em número de pixels, da imagem.

Os momentos de primeira ordem ($(p,q) = (1,0)$ e $(p,q) = (0,1)$) contêm informação sobre o centro de gravidade do objeto, e são determinados pelas equações 4.5 e 4.6.

$$m_{1,0} = \sum_x \sum_y xI(x,y) \tag{4.5}$$

$$m_{0,1} = \sum_x \sum_y yI(x,y) \tag{4.6}$$

Os momentos de segunda ordem ($(p,q) = (2,0)$, $(0,2)$ e $(2,2)$), por sua vez, contêm informação sobre os momentos de inércia em relação ao eixo vertical, horizontal e ambos. Estes são determinados pelas equações 4.7 a 4.9.

$$m_{2,0} = \sum_x \sum_y x^2 I(x,y) \tag{4.7}$$

$$m_{0,2} = \sum_x \sum_y y^2 I(x,y) \tag{4.8}$$

$$m_{2,2} = \sum_x \sum_y x^2 y^2 I(x,y) \tag{4.9}$$

Por seu lado, os momentos de terceira ordem ($(p,q) = (3,0)$ e $(0,3)$) estabelecem o grau de simetria do objeto em torno dos eixos vertical e horizontal, respetivamente.

Uma vez que $I(x,y)$ representa a intensidade do pixel (x,y), se a imagem sofre uma translação o valor da $m_{p,q}$ será também alterado. A fim de tornar $m_{p,q}$ invariante relativamente à translação de I, usam-se os momentos centrais. Estes tomam os mesmos significados que os momentos já descritos exceto pelo facto de x e y usados na fórmula serem deslocados pelo centróide da imagem. O momento central $\mu_{p,q}$ é calculado pela equação 4.10:

$$\mu_{p,q} = \sum_{i=0}^{n} I(x,y)(x-x')^p (y-y')^q \quad , \tag{4.10}$$

onde $x' = m_{1,0}/m_{0,0}$ e $y' = m_{0,1}/m_{0,0}$.

O momento normalizado, $\eta_{p,q}$ (equação 4.11), tem o mesmo significado do momento central, mas de forma a que o momento normalizado resultante seja invariante à escala do objeto. Assim, os momentos centrais são divididos por uma potência de $m_{0,0}$.

$$\eta_{p,q} = \frac{\mu_{p,q}}{m_{0,0}^{(p+q)/2+1}}. \tag{4.11}$$

4.2.2.3 Momentos invariantes de Hu

É possível combinar momentos centrais de forma a que estes se tornem invariantes em relação a translação, rotação e variações de escala. Ming-Kuei Hu apresentou em 1962 uma combinação linear de momentos invariantes para figuras geométricas planas [Hu62], aos quais damos o nome de momentos invariantes de Hu. Os 7 momentos invariantes de Hu são determinados pelas

expressões 4.12 a 4.18:

$$h_1 = \eta_{2,0} + \eta_{0,2} \tag{4.12}$$

$$h_2 = (\eta_{2,0} - \eta_{0,2})^2 + 4\eta_{1,1}^2 \tag{4.13}$$

$$h_3 = (\eta_{3,0} - 3\eta_{1,2})^2 + (\eta_{2,1} - \eta_{0,3})^2 \tag{4.14}$$

$$h_4 = (\eta_{3,0} - \eta_{1,2})^2 + (\eta_{2,1} + \eta_{0,3})^2 \tag{4.15}$$

$$\begin{aligned} h_5 = {} & (\eta_{3,0} - 3\eta_{1,2})(\eta_{3,0} + \eta_{1,2})\left((\eta_{3,0} + \eta_{1,2})^2 - 3(\eta_{2,1} + \eta_{0,3})^2\right) \\ & + (3\eta_{2,1} - \eta_{0,3})(\eta_{2,1} + \eta_{0,3})\left(3(\eta_{3,0} + \eta_{1,2})^2 - (\eta_{2,1} - \eta_{0,3})^2\right) \end{aligned} \tag{4.16}$$

$$\begin{aligned} h_6 = {} & (\eta_{2,0} - \eta 0,2)\left((\eta_{3,0} + \eta_{1,2})^2 - (\eta_{2,1} + \eta_{0,3})^2\right) \\ & + 4\eta_{1,1}(\eta_{3,0} + \eta_{1,2})(\eta_{2,1} + \eta_{0,3}) \end{aligned} \tag{4.17}$$

$$\begin{aligned} h_7 = {} & (3\eta_{2,1} - \eta_{0,3})(\eta_{2,1} + \eta_{0,3})\left(3(\eta_{3,0} + \eta_{1,2})^2 - (\eta_{2,1} + \eta_{0,3})^2\right) \\ & - (\eta_{3,0} - 3\eta_{1,2})(\eta_{2,1} + \eta_{0,3})\left(3(\eta_{3,0} + \eta_{1,2})^2 - (\eta_{2,1} + \eta_{0,3})^2\right) \end{aligned} \tag{4.18}$$

O primeiro momento de Hu tem um significado análogo ao momento de inércia em torno do centroide da imagem, sendo a intensidade de cada pixel análoga à densidade física. O último momento, h_7, é invariante a deformações laterais ou de inclinação.

4.2.2.4 Análise de componentes principais

Uma forma eficaz de conseguir uma boa representação da orientação de um objeto é através da análise de componentes principais (em inglês, *Principal Component Analysis* – PCA). Esta abordagem passa por encontrar o centro de um conjunto de amostras e, de seguida, encontrar os eixos principais desse grupo.

O primeiro eixo principal é aquele que passa pelo ponto médio do conjunto de amostras e que devolve a maior variância quando as amostras são projetadas neste. O segundo eixo é aquele que maximiza a variância numa direção normal à do primeiro eixo [Dav96]. Este processo é repetido até que N eixos principais sejam encontrados para um espaço com N dimensões.

O processo é ilustrado na figura 4.12, onde os pontos representam amostras no espaço, inicialmente medidas em relação aos eixos x e y. O ponto $0'$ representa o centro do conjunto de amostras. A direção $0'x'$da primeira componente principal maximiza a variância e a direção $0'y'$ da segunda componente principal é normal à direção $0'x'$.

Este processo é inteiramente matemático e resume-se a procurar um conjunto de eixos ortogonais que diagonalizem a matriz de covariância.

A matriz de covariância C para a população de entrada é definida pela equação 4.19:

$$C = E\left\{(x_{(p)} - m)(x_{(p)} - m)^\top\right\} \quad , \tag{4.19}$$

onde $x_{(p)}$ representa a localização da amostra p, m é a média do conjunto de pontos P e $E\{\dots\}$ indica o valor esperado da população subjacente. C pode ser estimado a partir das equações 4.20

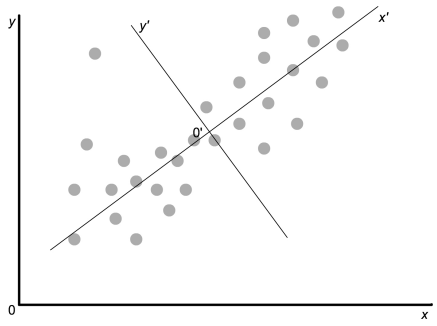


Figura 4.12: Ilustração da análise de componentes principais. Adaptado de [Dav96].

e 4.21.

$$C = \frac{1}{P}\sum_{p=1}^{P} x_{(p)}x_{(p)}^{\top} - mm^{\top} \tag{4.20}$$

$$m = \frac{1}{P}\sum_{p=1}^{P} x_{(p)} \tag{4.21}$$

Como a matriz C é real e simétrica, pode ser diagonalizada usando uma matriz de transformação ortogonal A (lembra-se que para uma matriz ortogonal $A^{-1} = A^{\top}$). Obtém-se portanto um conjunto de N vetores próprios ortonormais, u_i, com valores próprios λ_i, dados por $Cu_i = \lambda_i u_i$, onde $i = 1,2,\ldots,N$ e u é um vetor coluna. Os vetores u_i são derivados a partir dos vetores originais x_i, fazendo $u_i = A(x_i - m)$.

Pode ser demonstrado que as linhas da matriz A são formadas pelos vetores próprios de C e que a matriz de covariancia diagonalizada é dada por $C' = ACA^{\top}$, tal que:

$$C' = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \vdots \\ 0 & & \cdots & \lambda_N \end{bmatrix} \tag{4.22}$$

Se considerarmos que os valores próprios foram colocados em sequência, começando pelo mais alto, temos que λ_1 representa a característica mais relevante do conjunto de amostras e a relevância da informação contida nos valores próprios seguintes é sucessivamente menor. Assim, para um N suficientemente elevado os últimos valores próprios representam frequentemente características que são estatisticamente irrelevantes. É por esta razão que a análise de componentes principais é usada como uma forma de reduzir a dimensão de um espaço de dimensão N para um valor inferior, N', resultando numa redução significativa da redundância presente nas amostras do conjunto inicial.

O interesse em utilizar análise PCA está no fato desta ser uma forma rápida e eficiente de obter a orientação principal do objeto na imagem. Para o efeito, primeiro converte-se cada pixel da imagem para um vetor com origem no canto superior esquerdo da imagem, resultando numa matriz de duas colunas e N linhas, sendo N o número de pixels que descrevem a mão. Segue-se o

levantamento dos dois primeiros vetores principais u_1 e u_2. Finalmente, calcula-se o ângulo θ de cada um destes vetores. Obtemos assim dois ângulos que traduzem a orientação principal segundo os eixos x e y da mão.

4.2.3 Classificadores

Associada à ideia de reconhecimento de padrões está o conceito de "aprendizagem" a partir de amostras [GW02]: é necessário que um sistema tenha o conhecimento dos padrões que deverá reconhecer. Consideramos um padrão como uma junção de características como as discutidas na secção 4.2.2, enquanto que uma classe traduz uma família de padrões que contêm as mesmas características. O reconhecimento de padrões através de aprendizagem por computador (*machine learning*) envolve técnicas que sejam capazes de associar um determinado padrão a uma determinada classe de uma forma automática e com pouca interação humana. A estes métodos damos o nome de classificadores.

Neste trabalho, o nosso objetivo é reconhecer ortografia gestual, assim sendo as nossas classes correspondem às letras do alfabeto. Foram cuidadosamente escolhidas 4 letras de todo o conjunto (ver figura 2.3): as letras **A**, **L**, **O** e **T** e a sua morfologia é apresentada na figura 4.13.

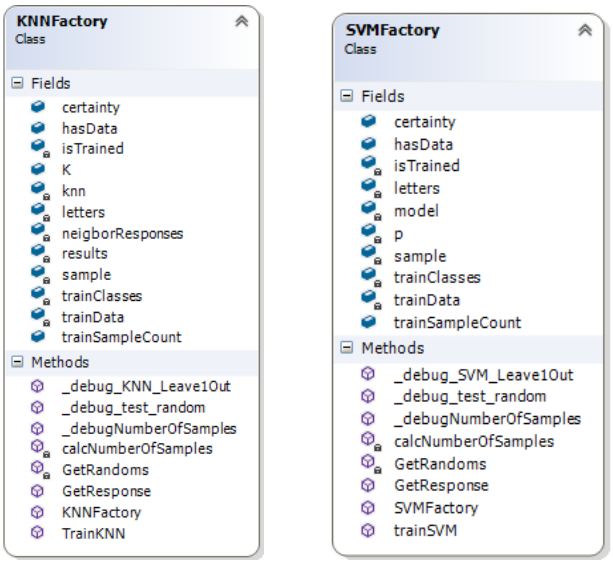
(a) Letra **A**. (b) Letra **L**. (c) Letra **O**. (d) Letra **T**.

Figura 4.13: Classes de padrões escolhidas para o sistema de reconhecimento.

Cada uma das classes é descrita por um vetor coluna de 9 caraterísticas, correspondentes aos 7 momentos invariantes de Hu e aos dois ângulos das duas primeiras componentes principais. Apesar da análise dos contornos e defeitos de convexidade terem sido inicialmente consideradas como caraterísticas a serem utilizadas neste vetor, a sua implementação lado-a-lado com as restantes mostrou-se problemática, tendo sido descartadas para o sistema final.

Note-se que procurou-se escolher classes com alguma variabilidade na sua forma, mas também algumas suficientemente próximas. As classes **L** e **T** são na sua essência o mesmo padrão, mas rodado de 90º. A escolha destas duas classes foi propositada com o objetivo de testar a eficácia das caraterísticas utilizadas para este tipo de casos.

Dos vários tipos de classificadores disponíveis na área de *machine learning* foram escolhidos e implementados dois algoritmos estatísticos de reconhecimento de padrões: *K-Nearest Neighbours* (K-NN) e *Support Vector Machine* (SVM). As seguintes secções descrevem o funcionamento destes. Para cada classificador foi criada uma classe que gere o seu funcionamento interno: inicialização, treino, classificação de amostras e testes de viabilidade, sendo os diagramas destas apresentados na figura 4.14.



(a) Classe *KNNFactory*. (b) Classe *SVMFactory*.

Figura 4.14: Diagramas da classes implementadas para supervisionar a operação dos classificadores K-NN e SVM.

O *Open CV* contém uma biblioteca direcionada para *machine learning*, que é encapsulada e acessível em C# através de *EmguCV*. Os algoritmos de classificação utilizados tiram partido das capacidades desta biblioteca.

4.2.3.1 *K-Nearest Neighbours*

Um dos classificadores mais simples é o de "k-vizinhos mais próximos" (em inglês *K-Nearest Neighbours*, K-NN). Este é um método não paramétrico de classificação, o que quer dizer que não utiliza nenhum conhecimento a priori sobre a natureza da distribuição das amostras. O classificador armazena os dados de todas as amostras de treino que tem disponíveis, ao conjunto das quais daremos o nome de *dataset*. Para classificar uma nova amostra, são procurados as K amostras mais próximas, sendo K um inteiro. Das K amostras mais próximas é contado o número de vezes que ocorre cada classe. A classe com maior incidência é provavelmente a classe da nova amostra e portanto esta é-lhe atribuída.

Para encontrar as amostras mais próximas podem ser usadas várias métricas de distância. Foi usada a distância euclidiana entre 2 pontos no espaço. O valor K é definido pelo programa.

Para treinar o classificador K-NN, são levantadas amostras que são classificadas pelo utilizador, fazendo deste um método de aprendizagem supervisionada. O processo de treino consiste em guardar os vetores de características para cada amostra e suas respetivas classes.

Uma das vantagens de usar o classificador K-NN, para além da sua simplicidade, é o facto deste, para um conjunto ótimo de amostras de treino, apresentar uma taxa de erro muito próxima da do classificador de Bayes (que não será abordado neste trabalho). No entanto, tem a desvantagem de necessitar, geralmente, de um *dataset* com um elevado número de amostras/classe precisando

de um poder de armazenamento considerável para preservar todas estas e, correspondentemente, um elevado poder computacional para achar a aproximação ótima para cada amostra. No entanto, o uso de vetores de características de dimensões reduzidas como os utilizados neste projeto reduz em muito a complexidade computacional do sistema.

4.2.3.2 *Support Vector Machine*

SVM é mais um método de aprendizagem supervisionada. No seu conceito mais elementar consiste em encontrar um par de hiperplanos paralelos que levam à máxima separação entre duas classes de características, de forma a assegurar a maior proteção contra erros [Dav96].

A figura 4.15 ilustra o conceito básico. Esta apresenta duas classes linearmente separáveis. Os dois hiperplanos paralelos apresentam a maior separação possível, d e devem ser comparados com alternativas como os hiperplanos a tracejado. Estes últimos apresentam menor separação entre amostras, resultando em menor proteção contra erros.

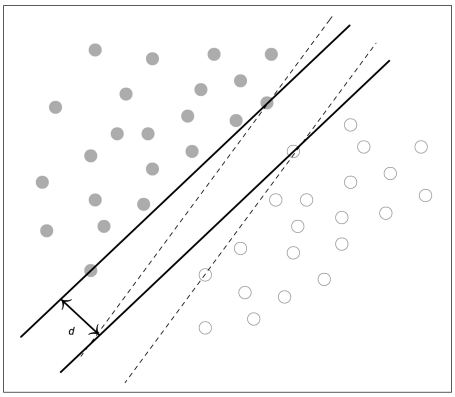


Figura 4.15: Representação do principio de SVM. Adaptado de [Dav96].

Cada par de hiperplanos paralelos é caracterizado por um conjunto específico de características às quais se dá o nome de vetores de suporte (*support vectors*). No espaço de características da imagem 4.15, os planos podem ser totalmente definidos por três vetores de suporte, o que acontece apenas para um espaço de características bidimensional. Tendo um espaço de dimensão N, o número de vetores de suporte necessários é de $N+1$.

Para um número de classes N o algoritmo funciona ao projetar as amostras num espaço de dimensão superior, criando novas dimensões da combinação de características. Por fim, procura-se o melhor conjunto de separadores entre as classes.

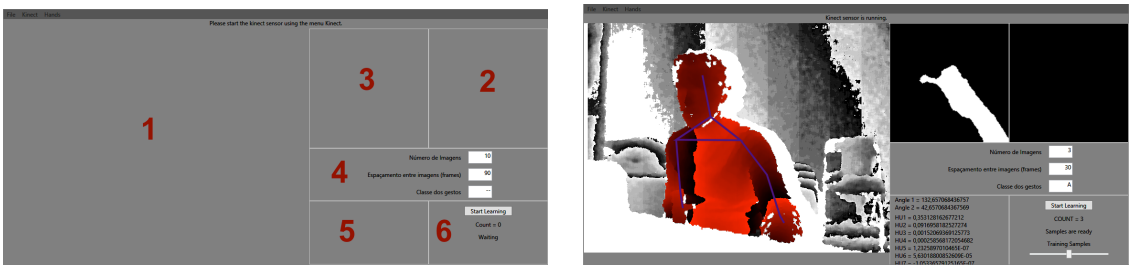
No espaço de amostras de entrada o classificador linear de elevada dimensão pode-se tornar bastante não linear [BK08]. Portanto podem-se empregar técnicas de classificação baseadas na máxima separação inter-classes para produzir classificadores não lineares que melhor consigam separar as classes das amostras.

Utilizando a biblioteca de *Machine Learning* do *EmguCV* a implementação de um classificador SVM torna-se simples. A biblioteca permite treino automático do classificador, ao procurar

os melhores parâmetros para um determinado *dataset* e, assim, o classificador procura o melhor conjunto de hiperplanos que garantem a melhor proteção contra erros possível.

4.2.4 Sistema de treino

Como os dois classificadores implementados são da categoria de aprendizagem supervisionada, foi necessário arranjar uma forma de criar um conjunto de amostras que conseguisse, de uma forma fácil e eficiente, representar os padrões de cada classe, isto é gesto. Para o efeito modelou-se um sistema de treino capaz de extrair características em tempo real e armazenar estas de uma forma eficiente.



(a) Antes de iniciar a captura de amostras. (b) Após captura de amostras.

Figura 4.16: Interface do sistema de treino.

Na figura 4.16 apresenta-se a interface da aplicação que foi desenvolvida para o levantamento e classificação de amostras de treino, esta é composta por 6 regiões, devidamente assinaladas na figura 4.16a:

1. Resultado do pré-processamento do *Depth Stream* e *Skeleton Stream*; apresenta-se como uma forma do utilizador compreender se está a ser detetado e o seu posicionamento, em tempo real;

2. Resultado da segmentação da mão em tempo real;

3. Local onde surge a última amostra processada;

4. Configurações de captura:

   - Número de amostras a capturar;
   - Espaço entre amostras, em frames/segundo;
   - Classe do conjunto de amostras a ser capturado;

5. Características da amostra apresentada na zona 3;

6. Zona de informação. Contem informação sobre estado do processo de captura de amostras a decorrer.

Antes de iniciar a captura de um conjunto de amostras o utilizador deve configurar o número de amostras a levantar pelo sistema e o espaçamento entre estas. Como o sistema opera a $30fps$ um valor de 30 neste parâmetro equivale a 1 segundo entre amostras capturadas, o que permite ao utilizador reposicionar a mão de forma a obter diferentes variações do mesmo gesto.

Finalmente, é necessário configurar a classe dos gestos. Cada sessão de captura deverá referir-se a uma só classe. No fim do processo, o utilizador deverá comandar a aplicação para guardar os dados relativos às amostras e será usada a classe definida para todas as amostras da sessão corrente. A classe das amostras reflete a letra correspondente ao gesto e deve ser representada em letra maiúscula.

Uma vez completo o processo de captura na zona 6 é desligada a apresentação do resultado em tempo real (zona 2) e surge um *slider* que permite percorrer todas as amostras da sessão. A informação relativa a cada amostra é então apresentada na zona 5 e a respetiva forma da mão em 3.

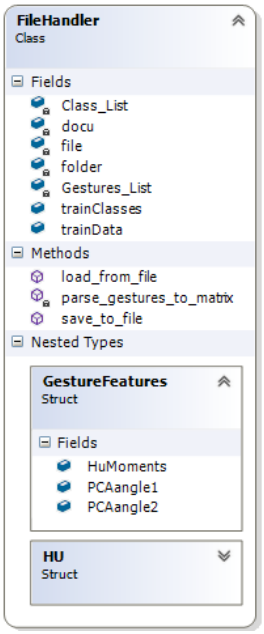


Figura 4.17: Classe de gestão de escrita e leitura de informação de características para um ficheiro XML.

Se o utilizador achar que todas as amostras estão corretas estas podem ser armazenadas através da barra de menu. As amostras são guardadas num ficheiro XML a fim de assegurar persistência dos dados. Para gerir o processo de passagem de informação para o ficheiro e, posteriormente, aceder a este, foi criada uma classe *FileHandler* cuja estrutura é apresentado na figura 4.17.

O sistema permite, ainda, a captura de amostras com ambas as mãos. Para alternar a mão a ser detetada pelo processo de segmentação o utilizador deverá escolhe-la através do barra de menu.

As amostras são sempre guardadas no mesmo ficheiro *TrainedClasses.xml* para que o utilizador possa gerar várias sessões com diferentes classes aumentando assim o *dataset*.

Com este sistema de treino foram criados alguns conjuntos de amostras, incluindo vários conjuntos com amostras de uma só pessoa com 50, 100, 150, 200, 250 e 300 amostras/gestos e um outro com 50 amostras/gesto num total de 4 pessoas.

A fim de assegurar pouca variação na qualidade das amostras levantadas foi definido um pequeno protocolo para o processo de recolha de amostras de treino que se limita a duas premissas:

1. O ângulo de elevação do sensor Kinect deve-se encontrar dentro da gama $[5^o; 10^o]$;
2. O sujeito deve se encontrar uma distância do sensor entre $0.7m$ e $1m$.

Para testar a viabilidade de um *dataset*, foram implementadas para cada classificador dois métodos que devolvem métricas de análise deste. Entenda-se como viabilidade de um conjunto de amostras como a taxa de deteção correta que um classificador apresenta quando treinado com esse mesmo conjunto de amostras.

O primeiro método usa um número de amostras aleatórias, dado pelo utilizador, do conjunto total. A este conjunto damos o nome de série de teste. Usam-se as restantes amostras (série de treino) para treinar o classificador. A viabilidade do *dataset* corresponde à percentagem de amostras da série de teste corretamente classificadas. Como as amostras da série de treino são escolhidas aleatoriamente, esta métrica de avaliação não devolve valores constantes e o facto de não usar a totalidade do *dataset* para treinar o sistema, vem também piorar a avaliação deste como um todo.

Assim, desenvolveu-se um segundo método para testar a viabilidade do *dataset*. Este baseia-se no algoritmo *Leave 1 Out* que se baseia em iterar todas as amostras do conjunto. A amostra corrente corresponde à série de teste, as amostras restantes são as séries de treino. Assim, para cada amostra, o classificador é treinado com as restantes e testado com esta. Quando todas as amostras foram testadas, a viabilidade representa o número de amostras que foram corretamente classificadas em todo o conjunto. O resultado deste algoritmo para um determinado *dataset* será sempre constante.

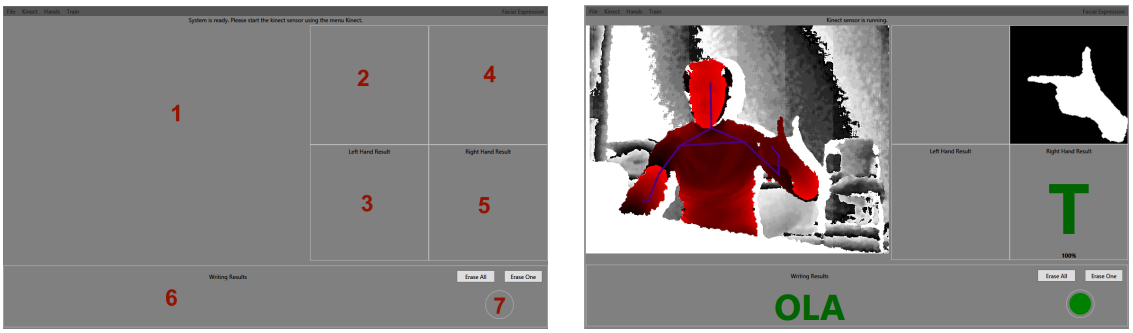
4.2.5 Ambiente de deteção

Tendo levantado as características de cada gesto, criado classificadores para o reconhecimento de padrões e um conjunto de amostras que serviram para treinar estes classificadores, passou-se a criar o sistema de deteção e reconhecimento de elementos da ortografia gestual.

A interface da aplicação implementada é apresentada na figura 4.18. É também esta a interface que confere acesso às aplicações de treino e de reconhecimento da expressão facial, previamente apresentadas.

A interface da aplicação é composta por 7 regiões principais devidamente anotadas na figura 4.18a. A região 1, como na interface do sistema de treino, apresenta o resultado do processo de pré-processamento do *Depth Stream*, adicionando a informação relativa ao rastreio das articulações do esqueleto do utilizador do *Skeleton Stream*.

O sistema é capaz de detetar e seguir a mão esquerda, direita ou as duas em simultâneo. Para escolher a mão a seguir, o utilizador deverá ativar o rastreio da devida articulação usando o



(a) Antes de iniciar o sensor e reconhecimento.

(b) Durante o reconhecimento de gestos.

Figura 4.18: Interface do sistema de deteção.

menu *Hands* na barra de menu. Esta capacidade do sistema foi implementada devido ao fato de a ortografia gestual não ser praticada com uma mão em específico. Esta prática depende de qual é a mão principal do utilizador, isto é, depende do utilizador ser destro ou canhoto.

As zonas 2 e 4 apresentam a informação relativa ao processo de segmentação da mão esquerda e direita, respetivamente. As regiões 3 e 5 apresentam o resultado da classificação por parte do classificador, ou seja, a letra detetada, para a mão correspondente.

No caso de se usar o classificador K-NN, foi definido um grau de certeza da resposta do classificador. Como o algoritmo procura as K amostras mais próximas e devolve a classe que ocorre mais vezes nesse conjunto, definiu-se o grau de certeza C como a percentagem de amostras, em K, cuja classe corresponde à resposta do classificador. Com este grau de certeza, modela-se também a cor da letra que surge nas regiões 3 e 5. Assim damos ao utilizador uma forma de, visualmente, se assegurar da robustez da deteção. Como o classificador SVM opera de uma forma distinta, este cálculo não é possível. Neste caso, a cor da letra detetada é constante.

Se um gesto for mantido por um segundo sem alteração, isto é, se nenhum padrão diferente for detetado, a letra exposta em 5 (ou 3) é passada para a região 6. Esta região funciona como uma área de escrita, oferecendo a possibilidade de escrever palavras através de ortografia gestual, como se pode ver na figura 4.18b, quando se escreveu a palavra “olá”.

Finalmente, a região 7 opera como um marcador de tempo. Quando um gesto é detetado o círculo começa a ser preenchido com a cor verde. Se o gesto for mantido por 1 segundo, este será completamente preenchido, momento no qual a letra detetada é passada para a região 6 e o círculo volta ao seu estado neutro, ou vazio.

4.3 Sumário

Neste capítulo abordou-se todo o desenvolvimento realizado para atingir a deteção tanto de expressão facial como de ortografia gestual.

Para atingir a deteção da expressão facial começámos por estudar o pacote de desenvolvimento *Face Tracking SDK*, que acrescenta ao KinectSDK a capacidade de detetar a morfologia da cara do utilizador. Vimos que as unidades de animação, popularmente utilizadas através do sistema

FACS, nos concedem uma forma de analisar expressões faciais simples em tempo real. Estudámos o modelo de parametrização da face CANDIDE e como este é utilizado pelo *Face Tracking SDK*.

Finalmente, descreveu-se o procedimento utilizado para detetar expressões faciais utilizadas na LGP através da parametrização de expressões faciais, utilizando unidades de animação e apresentou-se a interface de reconhecimento desenvolvida. Utilizando duas expressões básicas demonstrou-se que o sistema é capaz de reconhecer expressões faciais simples em tempo real, de acordo com os objetivos propostos.

Passou-se, então, ao estudo do reconhecimento de ortografia gestual. Iniciou-se este estudo com o processo de pré-processamento da informação captada pelo sensor Kinect. Este processo passa primeiro por detetar a posição da mão do utilizador, no espaço da informação captada pelo sensor. A fim de conseguirmos ter melhor informação sobre a morfologia da mão, é empregue um processo de segmentação da área de interesse que se baseia num método de limiarização da informação de profundidade.

Para que o sistema possa inferir sobre a pose da mão é necessário extrair informação relevante da imagem segmentada. Vimos que os contornos e pose da mão podem ser obtidos pelo estudo da imagem através da análise do seu envelope convexo, juntamente com os defeitos de convexidade. Como esta informação não é suficiente para um sistema robusto, verificámos que o cálculo dos momentos do objeto nos concede características do seu contorno que são invariantes relativamente a translação, rotação e variações de escala.

Uma outra característica estudada foi a análise das componentes principais de uma imagem. Vimos que esta é uma forma rápida e eficiente de obtermos informação sobre a orientação da mão no espaço.

O levantamento de caraterísticas de uma imagem não é suficiente para que um sistema consiga reconhecer um gesto. No entanto, estas formam padrões distintos para cada gesto. Estudaram-se então classificadores que são capazes de aprender classes de padrões e associar novas amostras a estas. Estudamos dois algoritmos de aprendizagem supervisionada preparados para o efeito, o K-NN e o SVM.

Algoritmos de aprendizagem supervisionada, como os dois utilizados, requerem a classificação de um conjunto de amostras para que estas possam ser usadas para treinar os classificadores. Foi então desenvolvido um sistema de treino com a capacidade de levantar uma série de amostras e armazená-las para que possam ser corretamente utilizadas aquando da operação de deteção.

Finalmente, abordámos a aplicação de reconhecimento. Foram escolhidas quatro letras do alfabeto gestual: **A**,**L**, **O** e **T** para demonstrar o seu funcionamento. São usados os momentos invariantes de Hu e os dois primeiros ângulos principais de cada amostra para criar vetores de 9 caraterísticas que traduzem o padrão de cada gesto. O vetor é então passado ao classificador, devidamente treinado, a fim de aferir a classe do gesto a ser efetuado. O sistema opera a $30 fps$ conseguindo reconhecimento em tempo real.

Capítulo 5

Análise de resultados

Associado ao desenvolvimento de uma aplicação surge normalmente uma análise descritiva do seu comportamento. Nas seguintes secções passamos a fazer essa análise para as aplicações de deteção da expressão facial e de gestos estáticos.

Começamos por abordar o sistema de reconhecimento da expressão facial. Procura-se investigar o desempenho deste na deteção das expressões faciais modeladas: expressão facial exclamativa e a expressão facial alegre. Para o efeito aborda-se primeiro o comportamento em tempo real do sistema. De seguida reflete-se sobre o pacote de desenvolvimento *Face Tracking SDK* e por fim estuda-e a solução proposta através do uso das unidades de animação.

Passa-se então à análise do sistema de reconhecimento de gestos estáticos, começando por uma análise do processo de segmentação. Segue-se o estudo da eficiência das caraterísticas passando por observar o comportamento dos classificadores implementados que as utilizam. Por fim, analisa-se o sistema de reconhecimento de gestos estáticos, no que diz respeito ao seu desempenho em tempo real.

5.1 Reconhecimento da expressão facial

Uma vez produzida a aplicação de reconhecimento da expressão facial esta foi testada para avaliar a sua capacidade de efetivamente detetar a face do utilizador e a sua morfologia.

Muitos fatores contribuem para o bom funcionamento da aplicação. Entre estes destacam-se a qualidade do sensor Kinect e a própria morfologia da cara do utilizador. Nas seguintes secções comentam-se os resultados observados para a aplicação desenvolvida.

5.1.1 Deteção em tempo real

A qualidade da operação em tempo real é um dos fatores mais relevantes neste tipo de aplicações. Devemos relembrar que ao dizermos que uma aplicação opera em tempo real referimos-nos à capacidade desta processar informação à medida que ocorre, com uma latência entre a imagem captada e a apresentação do resultado do seu processamento quase impercetível.

No âmbito da Língua Gestual Portuguesa a expressão facial desenhada funciona, conforme referimos anteriormente, como um modificador do sentido. Assim sendo, este tipo de expressões são mais demoradas ocorrendo durante todo o tempo necessário para executar o gesto. Este facto contrapõe-se ao caso de uma expressão facial natural que pode ser quase instantânea, podendo ocorrer durante frações de segundo apenas. Não é, portanto, muito relevante a rapidez da deteção, mas sim a qualidade desta.

Para o *Face Tracking SDK* operar com boa viabilidade é preferível que se use o modo de alta resolução do *Color Stream* que limita a velocidade de processamento a apenas $12 fps$, com uma resolução de 1280×960 pixels. A aplicação de deteção estará então restringida a processar ao ritmo de captura do sensor. O número de imagens processadas por segundo afeta a experiência do utilizador diretamente. Este número, apesar de baixo, é suficientemente alto para dar a aparência de uma resposta em tempo real.

Como o *Color Stream* consegue também operar a $30 fps$, com uma resolução inferior de 640×480, testou-se a viabilidade da deteção com este modo a fim de verificar melhorias no processamento. O número menor de pixels afeta, porém, diretamente a resposta do *Face Tracking SDK*.

O nosso objetivo é detetar com acuidade a expressão. Assim, e considerando que uma expressão será mantida por pelo menos um segundo, é preferível abdicar do número de imagens processadas por segundo em prol de um melhor funcionamento.

5.1.2 *Face Tracking SDK*

O sistema de deteção e rastreio da cara através do *Face Tracking SDK* está apto a detetar efetivamente a face do utilizador no espaço da imagem. Não obstante a sua operação não é perfeita. Os testes realizados com o sistema de deteção da expressão facial permitiram comprovar que alguns aspetos da morfologia da face do utilizador podem afetar negativamente o funcionamento do sistema.

Como exemplo, se o sujeito a ser detetado utilizar óculos, o sistema é capaz de confundir a borda superior destes com as sobrancelhas. O resultado do seguimento das sobrancelhas é então mal processado e a respetiva unidade de animação tende a ser perto de constante. Outro caso observado resulta da existência de barba. Este erro na deteção é apresentado na figura 5.1. Se o bigode for escuro e bem marcado o sistema tende a confundi-lo como parte dos lábios. No caso da figura, o lábio superior é trocado com o bigode e o lábio inferior é interpretado como sendo o superior. O modelo da face apresenta-se, portanto, com a boca fechada numa posição neutra e o movimento do maxilar é registado de forma limitada ou, no pior dos casos, não apresenta nenhuma variação.

Outro fator limitativo é a redução, pelo *Face Tracking SDK*, do ângulo com que a cabeça do utilizador se deve apresentar para um bom reconhecimento (ver tabela 4.1). Esta limitação estende-se ao produto final. O utilizador não possui uma gama de movimento livre muito alta. Este aspeto pode ser combatido com um bom posicionamento do sensor. O sensor deve ser colocado

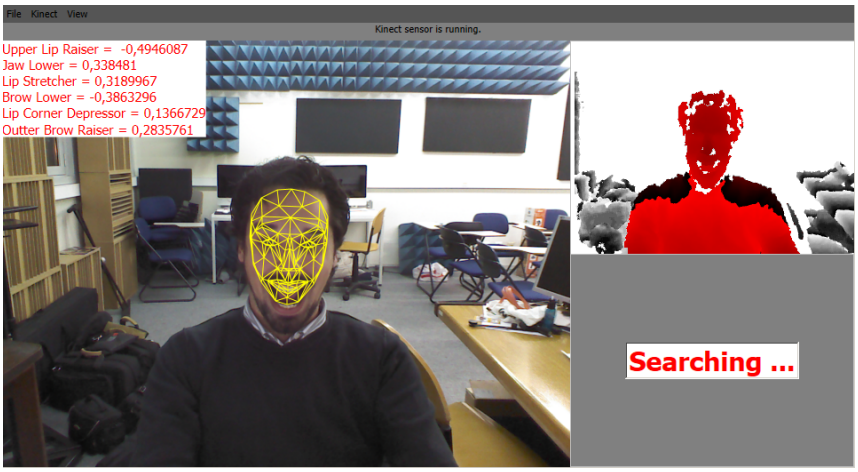


Figura 5.1: Deteção errada do estado da boca.

diretamente na frente do utilizador e o seu ângulo de inclinação deve-se encontra no intervalo $[-10^o; 10^o]$.

5.1.3 Unidades de animação

O modelo usado para atingir deteção das expressões faciais depende fortemente da análise e parametrização das unidades de animação disponíveis com o *Face Tracking SDK*. Em primeiro lugar, o número de unidades de animação disponíveis é, em si, uma fraqueza.

Como vimos na secção 4.1.3 estão disponíveis apenas 6 das 11 AUs do modelo CANDIDE. Esta redução do conjunto de unidades de animação impossibilita uma parametrização direta de todas as unidades existentes no modelo FACS. Uma demonstração deste facto pode ser observada na tabela 4.4: a expressão facial exclamativa é composta, no código FACS, por 4 diferentes unidades de animação, o *Face Tracking SDK* permite-nos modelar apenas 2 destas. Assim, a modelação de uma expressão facial pode ter que ser sujeita a algumas assunções sobre a sua forma.

Durante a análise das unidades de animação para cada expressão facial parametrizada notou-se que os valores destas não são muito estáveis. Se o utilizador mantiver, tanto quanto possível, a face numa determinada posição nota-se uma variação ligeira no cálculo das unidades de animação. Este efeito pode ser atribuído a pequenas variações dos músculos, assim como à pequena trepidação das imagens provenientes do Kinect, que é um efeito normal em qualquer sensor de captação de imagem. Ainda, notou-se que os valores de fronteira das AUs (ver tabela 4.2) representam extremos. Isto é, para certas unidades, por mais que se esforce o conjunto de músculos traduzidos por estas, raramente se obtêm os valores de 1 ou -1.

Este efeito ocorre porque as unidades de animação representam desvios das unidades de forma (SUs). Estas últimas são medidas no momento inicial de deteção e mantêm-se estáticas durante todo o processo. Se, no momento inicial, o utilizador tiver músculos da face contraídos, com um ligeiro desvio da posição neutra da sua face, as unidades de forma estarão também desviadas. Este erro é posteriormente carregado para o cálculo das unidades de animação.

Por outro lado, os limites das unidades de animação fazem certas assunções sobre a flexibilidade dos músculos da cara. Esta flexibilidade não é constante para todas as pessoas, e como exemplos podemos referir: a amplitude de abertura do maxilar de duas pessoas pode variar; a capacidade de subir as sobrancelhas é limitada pela gálea aponeurótica que é responsável pela elevação destas; a habilidade de forçar os cantos dos lábios para baixo depende da morfologia da boca. Assim, é de esperar que um determinado indivíduo não consiga obter todos os valores possíveis em todas as unidades de animação.

Tendo estes fatores em conta, para uma melhor eficiência é necessário que, no momento de inicialização do sistema, o utilizador mantenha uma expressão facial o mais neutra possível.

5.1.4 Apreciação global

Considerando toda a análise realizada nas secções precedentes, resta-nos analisar o comportamento do sistema de deteção da expressão facial como um todo.

Após vários testes ao sistema podemos afirmar que este consegue efetivamente detetar e reconhecer as expressões que foram escolhidas. A partir do momento em que se obtém a deteção de um utilizador e da sua face, o sistema consegue distinguir uma expressão exclamativa de uma expressão alegre ou neutra. O reconhecimento das expressões é rápido e obtido em tempo real, como pretendido.

Deve-se referir que o uso das três fontes de informação (*Color Stream*, *Depth Stream* e *Skeleton Stream*) do sensor Kinect consome alguma memória. Notou-se que $8GB$ de memória será um requisito mínimo para processar toda a informação sem atrasos significativos. Não existe mais nenhum requisito fora este e os demais aconselhados pelo produtor do sensor.

5.2 Reconhecimento de gestos estáticos

O desenvolvimento de uma aplicação de deteção e reconhecimento requer um certo trabalho de análise sobre as suas partes, a fim de validar o resultado final. No decorrer do projeto, todos os módulos que levam ao reconhecimento foram analisados a fim de assegurar a sua viabilidade e a sua capacidade de responder às necessidades do sistema final.

Começa-se por considerar o comportamento do *Skeleton Stream* do sensor, uma vez que é essencial ao bom funcionamento do processo de deteção e segmentação da mão. O sensor consegue efetivamente detetar e seguir, com boa acuidade, as articulações do utilizador, embora o posicionamento destas não seja perfeito.

O ponto no qual o sistema deteta a articulação da mão no espaço da imagem tende a variar, correndo desde a base da palma até ao início dos dedos. Se a variação se limitar ao interior da palma a segmentação não sofre alterações. No entanto, devido a movimentos rápidos ou falhas na deteção de uma ou mais articulações, o ponto da articulação pode ser mal deduzido, resultando em má segmentação.

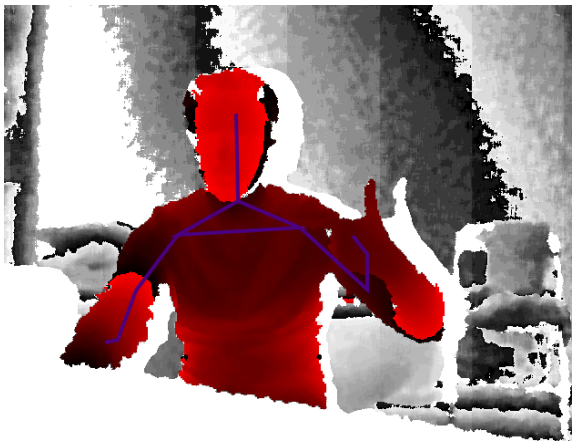


Figura 5.2: Rastreio das articulações do utilizador, note-se que devido à posição do braço o sistema não deteta bem a distância entre o cotovelo e o pulso.

Na imagem 5.2 vemos um exemplo no qual a articulação correspondente ao cotovelo do braço direito foi erroneamente posicionada a meio do braço, resultando em falha na deteção do antebraço. No entanto, o ponto da mão é corretamente detetado. Uma má deteção da mão não é habitual mas pode ocorrer principalmente se o utilizador se encontrar muito próximo do sensor. Como tal, aconselha-se uma distância mínima de 0.7 metros.

Como vimos, a segmentação da mão baseia-se num simples processo de limiarização. Este processo é eficiente uma vez que o seu resultado é correto e permite obter a forma da mão em diferentes configurações. Todavia, se a mão tocar em algum objeto, este passa a encontrar-se no mesmo limiar de profundidade, sendo a sua forma segmentada juntamente com a da mão. Neste caso perde-se informação, uma vez que o sistema não é capaz de distinguir a forma da mão quando esta surge ligada a um outro objeto. É necessário portanto que o utilizador não entre em contacto com nenhum objeto ou parte do corpo, como por exemplo a cabeça, durante a execução de um gesto para se obter a segmentação correta.

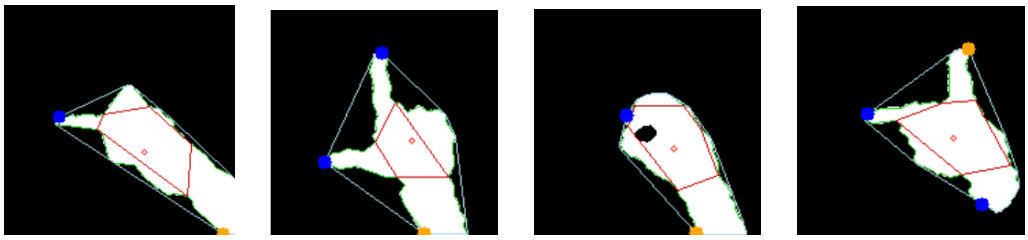
Nas seguintes secções descreve-se o resultado da análise efetuada aos restantes módulos do sistema. Começamos por examinar as características que foram implementadas, avaliando os seus potenciais e vulnerabilidades. Estas características são unidas de maneira a formarem padrões, que serão utilizados pelos classificadores para que o sistema seja capaz de reconhecer variados gestos. Assim, uma análise ao comportamento dos classificadores é também realizada.

5.2.1 Características

No processo de levantamento de características capazes de acrescentar conhecimento sobre a forma do gesto a ser efetuado, começámos por estudar o contorno do objeto e o seu envelope convexo, juntamente com a análise dos defeitos de convexidade.

Foi necessário arranjar métricas que pudessem dar-nos a capacidade de inferir sobre o número de dedos que o utilizador apresenta levantados, assim como a sua posição e tamanho relativo ao centro da palma da mão. Este método provou-se ser adequado ao cenário em que a mão está

posicionada verticalmente e apenas variam o número de dedos esticados. No entanto, os gestos utilizados na LGP não se baseiam somente na contagem dos dedos apresentados (ver figura 2.3). Assim, tivemos que analisar a eficácia de tal análise quando aplicada ao tema em mãos.



(a) Letra **A**. (b) Letra **L**. (c) Letra **O**. (d) Letra **T**.

Figura 5.3: Erros produzidos pela análise do envelope convexo para cada gesto considerado.

Na imagem 5.3 apresentam-se os resultados da análise do envelope convexo para os quatro gestos estudados. Lembra-se que nesta figura o ponto com maior distância ao centro da mão é representado a laranja, devendo identificar o maior dedo apresentado. A azul apresenta-se a ponta dos restantes dedos. Finalmente, a vermelho é representada a área convexa que engloba a palma da mão e um ponto que identifica o centro da palma.

Em primeiro lugar, temos o caso de sinais que não apresentam nenhum dedo levantado, como é o caso do gesto que corresponde à letra "O" (figura 5.3c). Para eliminar defeitos de convexidade que não acrescentam informação relevante, é necessário comparar os segmentos do envelope convexo preservando apenas aqueles que apresentam maior comprimento. No caso de letras que não expõem nenhum dedo levantado ("G", "O" e "S"), os segmentos do envelope convexo têm comprimentos curtos. O algoritmo considerará o maior destes. Assim, como podemos ver pela figura, o processo de extração pode considerar esses segmentos como relevantes, identificando-os efetivamente como dedos, gerando portanto informação enganosa.

Pode-se ver, também pela figura, um ponto detetado na base de todos os gestos (à esquerda). Este ponto é erroneamente classificado como um dedo. Além de aparecer em todos exemplos da imagem este não é constante, mas apresenta uma taxa de ocorrência muito elevada. A sua deteção deve-se à forma da base do objeto em si. O segmento de linha do envelope convexo que liga a este ponto é, geralmente grande, fazendo com que seja marcado como relevante pelo algoritmo.

Um processo capaz de remover este ponto depende do conhecimento do ângulo principal do objeto. Qualquer ponto que surja abaixo do centro da palma da mão não deverá ser considerado. No entanto, se considerarmos literalmente, todos os pontos da imagem que tenham uma coordenada y inferior ao fim da palma da mão corremos o risco de restringir a amplitude de movimentos do utilizador, o que vai contra os nossos objetivos.

Este problema foi o propulsor da análise das componentes principais efetuada. Como vimos, através desta análise conseguimos extrair os dois ângulos principais do objeto (segundo x e y). Esta seria informação suficiente para podermos deduzir em que região da imagem se encontra a base da mão, podendo, finalmente, passar a eliminar o ponto indesejado. Porém este não é o caso. É fácil de perceber que ao levantar apenas a primeira componente principal esta vai depender da

orientação do objeto analisado, não temos então forma de distinguir se esta se refere ao eixo x ou ao eixo y.

Os dois grandes problemas que este algoritmo apresenta tornam-no pouco viável pois não possui uma característica robusta que identifique o gesto de uma forma inequívoca. Assim, e como já vimos, passámos ao estudo de contornos através dos momentos da imagem.

As especificações da análise do comportamento dos classificadores cai no âmbito secção 5.2.2. No entanto, chamamos agora a atenção para o gráfico ilustrado na figura 5.4. Neste apresenta-se o comportamento do classificador K-NN quando se recorre a vetores de características que usam apenas os momentos invariantes de Hu.

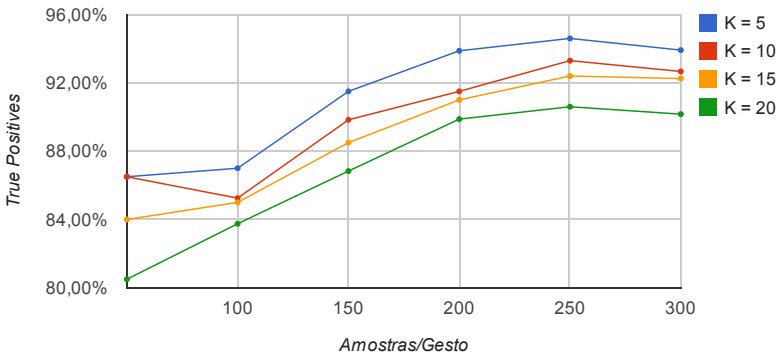


Figura 5.4: Taxas de viabilidade para o classificador K-NN. O vetor de caraterísticas usado contem apenas os 7 momentos de Hu.

Com o auxílio dos resultados da imagem podemos ver que com apenas os momentos invariantes temos um comportamento aceitável atingindo uma taxa de deteção correta, com o *dataset*, de 94.6% para 250 amostras por gesto, usando $K = 5$. No entanto, no pior caso, nestes testes obtém-se uma taxa de erro na deteção das letras "L" e "T" próxima dos 25%. Nota-se ainda, que em tempo real, o classificador tem alguma dificuldade em distinguir entre estes dois gestos.

Como tínhamos discutido na secção 4.2.2.3, a utilização dos 7 momentos invariantes de Hu permite detetar um objeto na imagem mesmo quando este é submetido a translação, rotação e variação de escala. Como estamos a analisar a forma que o gesto projeta, quando duas configurações são muito similares pode gerar-se confusão. As classes **L** e **T** são, essencialmente, o mesmo gesto rodado de 90^{o} para a direita. Precisamos então de mais uma característica que nos levante a incerteza entre gestos deste tipo. Mais uma vez, os dois ângulos principais foram resposta. A partir destes teremos, para cada gesto as suas componentes principais que ajudam a fazer a distinção entre duas formas parecidas com diferentes orientações.

Repetidos os testes que geraram o gráfico da figura 5.4, agora introduzindo no vetor de características os ângulos das duas componentes principais, obtemos os resultados observáveis na figura 5.5.

Nota-se que a viabilidade desceu consideravelmente, mas tende a diminuir com o aumento do número de amostras por gestos. Esta quebra na eficiência da deteção deve-se à introdução não

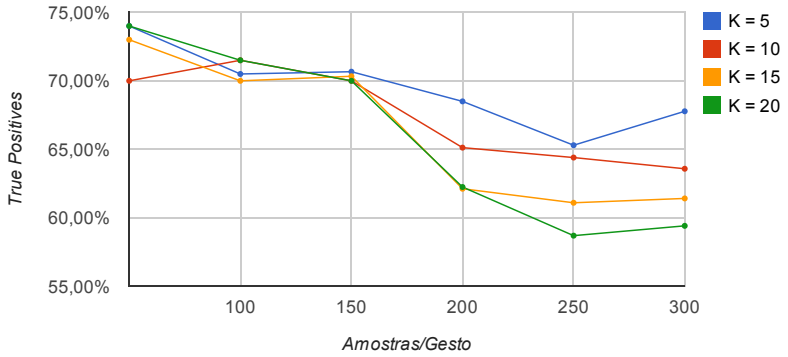


Figura 5.5: Efeito da variação do número de amostras/gesto no conjunto de treino e do valor de K no classificador K-NN.

normalizada dos ângulos que se encontram no intervalo $[-180^o; 180^o]$. Para percebermos este efeito temos que compreender o que está a acontecer ao vetor de características.

Na tabela 5.1 mostra-se o padrão que tomam os valores dos momentos invariantes para cada classe considerada no sistema. Pode ver-se que os valores dos momentos da imagem são muito baixos, muito próximos de 0. Assim, quando postos lado a lado com dois ângulo inteiros estes últimos terão muito peso no cálculo da distância entre 2 amostras. Será portanto mais importante a direção com que o gesto é desenhado do que a sua forma. Este não é o efeito desejado. É necessário normalizar os ângulos das componentes principais antes de os usar como características, a fim de diminuir a sua influência. Foram testadas várias gamas de normalização observando o efeito produzido na eficiência do classificador. Esta análise é discutida com mais pormenor na secção seguinte.

Tabela 5.1: Valores demonstrativos dos momentos de Hu para cada classe de gestos.

	h_1	h_2	h_3	h_4	h_5	h_6	h_7
A	0.32393	0.07553	0.00136	0.00015	$-4.64738E-8$	$-3.19190E-5$	$-5.08082E-8$
L	0.27807	0.02901	0.00453	0.00012	$8.79685E-8$	$1.03472E-6$	$-3.45127E-9$
O	0.26363	0.03765	0.00129	0.00026	$1.57677E-7$	$5.20144E-5$	$-8.31379E-9$
T	0.30705	0.04712	0.00511	0.00015	$1.26324E-7$	$2.96329E-5$	$-5.59878E-8$

5.2.2 Classificadores

Na secção anterior descrevemos o comportamento das caraterísticas quando aplicadas a um classificador para deteção de diferentes gestos. Nesta secção focamo-nos no comportamento dos classificadores. Variaram-se parâmetros das suas características, assim como do próprio conjunto de amostras que serve para os treinar.

Não é trivial quantificar a viabilidade de um classificador através de observações em tempo real. Muitos fatores podem perturbar os resultados, como o ângulo do sensor, a execução correta do gesto e variações na mão do utilizador, entre outros. Assim, tende-se a classificar a viabilidade

do conjunto das amostras de treino que o classificador vai usar: um bom conjunto de amostras formará, em principio, um classificador com boa viabilidade em tempo real.

Em primeiro lugar devemos assentar que um classificador só consegue inferir sobre a classe de uma determinada amostra relativamente aos padrões com que foi treinado. Assim, se um certo padrão não foi treinado, o classificador vai devolver uma resposta associando essa amostra a uma das classes conhecidas.

Analisou-se a viabilidade do sistema quando treinado com vários *datasets*. Como se pretende analisar o seu comportamento com o aumento do número de amostras por classe, os conjuntos de treino tiveram respetivamente 50, 100, 150, 200, 250 e 300 amostras por gesto. Estas amostras foram captadas com um só individuo.

Um algoritmo de deteção de padrões geralmente é avaliado pela sua capacidade de identificar corretamente as diferentes classes de padrões para o qual foi treinado. O cálculo da precisão de um classificador, ou a sua capacidade de detetar corretamente uma determinada classe, pode conduzir a uma análise enganosa. Se um determinado conjunto de amostras de treino contiver 95 amostras de uma classe $\mathbf{C}_1$ e apenas 5 amostras da classe $\mathbf{C}_2$, é fácil de perceber que o sistema vai ter uma tendência para a classe $\mathbf{C}_1$. Neste cenário, a precisão geral será de 95% mas, na prática, a taxa de deteção da classe $\mathbf{C}_1$ é de 100% e da classe $\mathbf{C}_2$ é de 0%. Assim, tende-se a usar quatro parâmetros para avaliar um classificador:

- *True Positives*
  Amostras da classe $\mathbf{C}_1$ que são detetadas como pertencendo à classe $\mathbf{C}_1$;

- *False Positives*
  Amostras da classe $\mathbf{C}_2$ que são erroneamente classificadas como da classe $\mathbf{C}_1$;

- *False Negatives*
  Amostras da classe $\mathbf{C}_1$ que são erroneamente classificadas como pertences à classe $\mathbf{C}_2$;

- *True Negatives*
  Todas as restantes amostras que foram corretamente classificadas como não pertencentes à classe $\mathbf{C}_1$.

Como o nosso objetivo é concretizar um sistema que detete corretamente os gestos, não nos interessa muito que o classificador marque um gesto **A** como **T**. Queremos a maior taxa de *true positives* possível. Assim, a rotina *Leave 1 Out*, criada para medir a viabilidade do conjunto de amostras de treino, retorna-nos apenas esta taxa de análise, juntamente como o número de classificações erradas por classe.

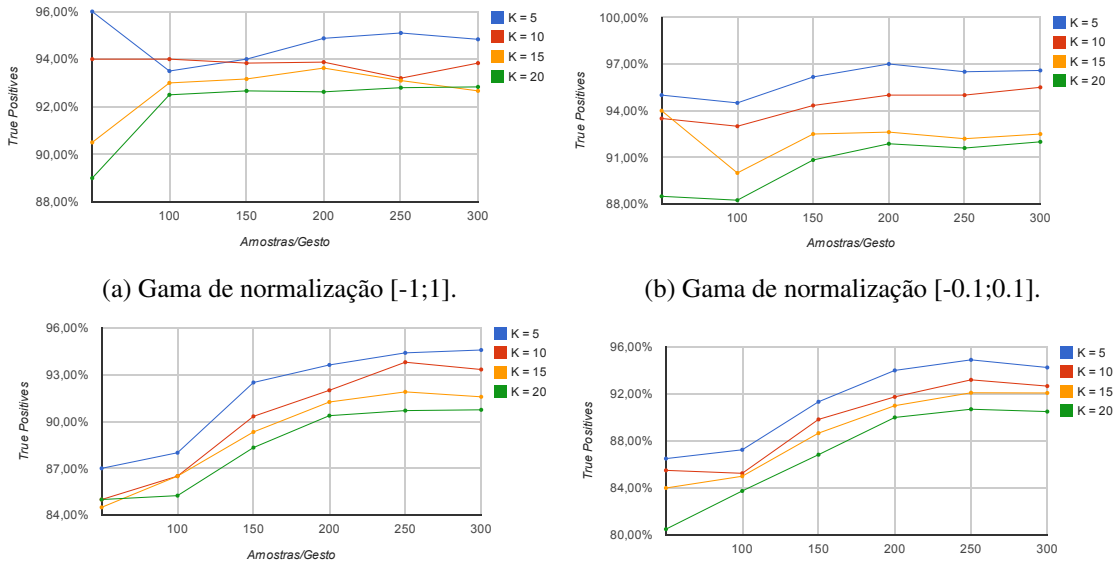
Toda a análise discutida sobre os classificadores é realizada, portanto, tendo como suporte o teste acima mencionado.

5.2.2.1 *K-Nearest Neighbours*

Como vimos em 5.2.1 houve a necessidade de normalizar a espaço de valores dos ângulos das componentes principais do objeto para que estes tivessem, no cálculo da distância euclidiana utilizada pelo K-NN, um peso igual (ou inferior) ao dos momentos invariantes.

Com o intuito de procurar a gama mais eficiente testaram-se variados espaços de normalização. Começou-se no intervalo $[-1;1]$ e prosseguiu-se dividindo este por um fator de 10. Na figura 5.6 apresentam-se os resultados deste exercício. Como anteriormente, aproveitam-se os mesmos gráficos para analisar o efeito da variação do valor de K no classificador, assim como a influência do número de amostras por gesto do conjunto de treino.

Podemos começar pela conclusão mais simples: o valor de K. Como vimos em 4.2.3.1, o algoritmo K-NN procura os K pontos, no espaço de amostras, mais próximos da entrada, devolvendo a classe que mais vezes aparece nestas K amostras mais próximas. O aumento deste valor permite ao classificador comparar mais amostras. A sua redução torna essa decisão mais restrita. Se observamos os gráficos das imagens 5.4, 5.5 e 5.6 constata-se que o algoritmo tende a oferecer melhores resultados para um valor de K mais baixo. Como o classificador é obrigado a considerar menos pontos, estes serão os que de mais perto se aproximam do padrão da entrada. Uma vez que a gama de valores dos momentos invariantes de Hu não é muito elevada (ver tabela 5.1), um ligeiro desvio da entrada pode levar a uma classe errada. Neste caso pode acontecer que a classe que surge mais vezes, nas K mais próximas, deixe de ser a correta sendo portanto preferível restringir o número de amostras analisadas.



(a) Gama de normalização [-1;1].

(b) Gama de normalização [-0.1;0.1].

(c) Gama de normalização [-0.01;0.01].

(d) Gama de normalização [-0.001;0.001].

Figura 5.6: Desempenho do classificador K-NN. Influência do valor de K, da normalização dos ângulos das componentes em diferentes intervalos e do número de amostras/gesto no conjunto de treino.

Outra conclusão que se pode tirar da análise direta dos gráficos ilustrados nas figuras 5.4, 5.5

e 5.6, é o efeito do aumento de número de amostras por gesto no conjunto de padrões de treino. Como seria de esperar, quanto maior for o número de amostras com que o classificador for treinado melhor será o comportamento do classificador, uma vez que terá à sua disposição mais amostras com as quais comparar a entrada. Este comportamento, porém, não se verifica no caso em que os ângulos das componentes principais do objeto não foram normalizados (figura 5.5). Neste caso, como vimos, são os ângulos que têm maior peso no vetor de caraterísticas. Como a maioria dos gestos têm orientações similares o classificador tende a considerá-las como pertencentes à mesma classe. Assim, neste caso, o comportamento do classificador é tanto pior quanto maior for o número de amostras disponíveis.

Sendo que o maior número de amostras juntamente com um valor de K baixo tende a gerar os melhores resultados, passou-se a testar os espaços de normalização dos ângulos das componentes principais usando um conjunto de treino de 300 amostras/gesto e $K = 5$. Na imagem 5.7 apresenta-se a cor roxa a taxa de *true positives* ("TP%") para cada gama e, em forma de gráfico de barras, a percentagem de erros/classe (*false negatives*) em cada uma. No gráfico, a amostra marcada como "NE" corresponde ao caso em que o vetor de características usado contém somente os 7 momentos invariantes de Hu.

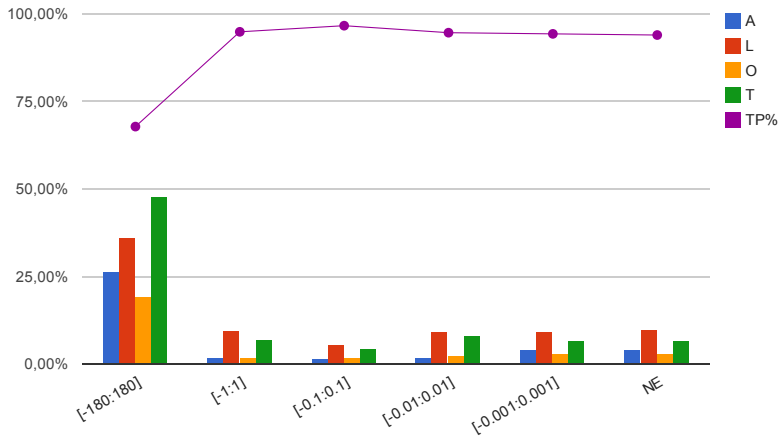


Figura 5.7: Comportamento do classificador K-NN para diferentes intervalos de normalização dos ângulos das 2 primeiras componentes principais. São usadas 300 amostras/gesto e $K = 5$.

Como podemos ver pela figura, a normalização dos ângulos das componentes principais não só aumenta consideravelmente a viabilidade do classificador mas também diminui a taxa de erros/letra. O melhor comportamento ocorre para a gama $[-0.1;0.1]$, onde se atinge 96.58% de viabilidade com o classificador. Apesar de superior ao caso em que apenas se usam os momentos da imagem (93.92%) o ganho de apenas cerca de 2.5% não parece muito significativo. No entanto, se observarmos as taxas de erro por letra, consegue-se notar a melhoria introduzida pelos ângulos.

Na figura 5.8, apresentam-se as taxas de erro por letra nas 3 situações onde se observou melhor desempenho. Podemos ver que a inclusão dos ângulos normalizados das componentes principais do objeto reduz efetivamente os erros na deteção, principalmente entre as classes **L** e **T**. Usando a gama $[-0.1;0.1]$ temos as menores taxas de erro de deteção, diminuindo esta quase para a metade.

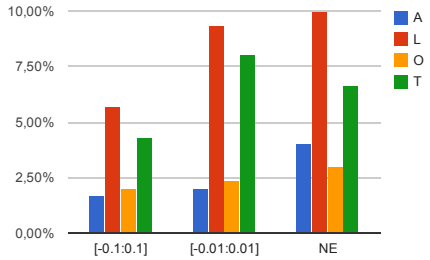


Figura 5.8: Taxas de erro/letra para os melhores intervalos obtidos com K-NN. São usadas 300 amostras/gesto e $K = 5$.

5.2.2.2 *Support Vector Machine*

Passamos agora a analisar o classificador SVM. Como vimos em 4.2.3.2, este algoritmo utiliza o conjunto de padrões de treino para criar hiperplanos que levem à melhor separação entre classes procurando reduzir o número de erros. Utilizando *Emgu CV*, o algoritmo adapta-se conforme as amostras de treino, procurando os melhores parâmetros para a maior eficiência.

Assim, para este classificador analisou-se apenas a influência do aumento do número de amostras e do intervalo de normalização dos ângulos correspondentes às duas primeiras componentes principais do objeto.

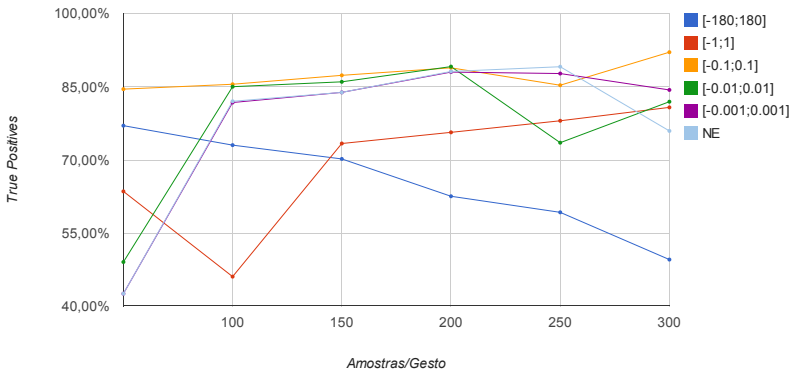


Figura 5.9: Taxas de viabilidade para o classificador SVM com a variação da gama de normalização dos ângulos das componentes principais

Na figura 5.9 apresentam-se os resultados dos testes realizados ao classificador. Na generalidade é observado um comportamento similar ao estudado com K-NN. O aumento do número de padrões de treino aumenta a eficiência do classificador, exceto, como já esperado, no caso em que os ângulos das componentes principais não estão normalizados. Neste classificador nota-se algumas quebras, ou descidas repentinas na taxa de *true positives*, o que se deve à parametrização automática do método implementado. Esta parametrização depende dos padrões de entrada e portanto são de esperar alguns erros.

A maior taxa de eficácia geral tende a encontrar-se no caso em que os *dataset* usado contem 200 amostras/gesto, onde se constatou uma taxa de *true positives* de cerca de 88% para vários

intervalos de normalização. Porém o melhor resultado global obtido regista 92.06% de viabilidade e ocorre para um conjunto de treino com 300 amostras/gesto e um intervalo de normalização dos ângulos de $[-0.1;0.1]$, equivalente ao obtido com o classificador K-NN.

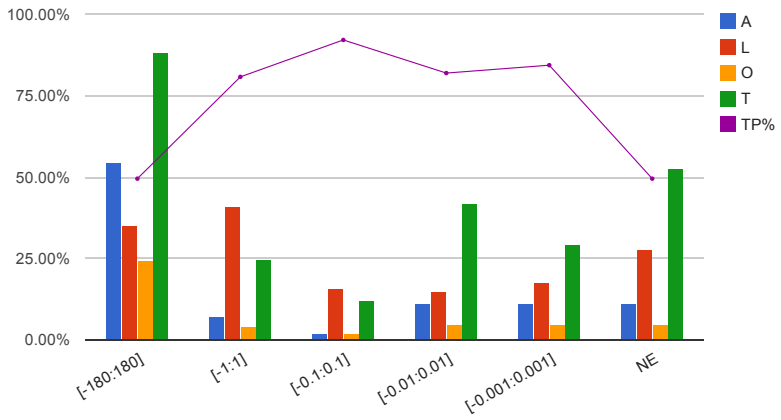


Figura 5.10: Taxas de erro/letra observadas com o classificador SVM para diferentes intervalos de normalização dos ângulos das 2 primeiras componentes principais. Foram usadas 300 amostras/gesto.

A figura 5.10 apresenta as taxas de erro por letra para o classificador SVM, quando o conjunto de amostras de treino é composto por 300 amostras/gesto. Verifica-se mais uma vez que para o intervalo $[-0.1;0.1]$ consegue-se obter a menor taxa de erros por letra.

Finalmente, constata-se que além do comportamento do classificador SVM ser similar ao do K-NN, o primeiro apresenta valores de viabilidade inferiores ao do último, apresentando-se portanto com pior desempenho geral.

5.2.3 Deteção em tempo real

A análise que fizemos ao sistema de deteção de gestos estáticos foi baseada nos resultados devolvidos pelo teste ao conjunto de amostras de treino. No entanto, é também relevante relatar as observações efetuadas durante a utilização em tempo real.

Como comprovado pelos testes ao *dataset*, o melhor comportamento em tempo real, para qualquer um dos classificadores utilizados, verifica-se no intervalo de normalização $[-0.1;0.1]$ para 300 amostras/gesto no conjunto de padrões de treino.

O sistema consegue detetar corretamente e sem erros as classes **A** e **O**; no entanto, e como seria de esperar pela análise precedente, apresenta por vezes dificuldade em diferenciar as classes **L** e **T** cometendo erros entre estas. Estes erros podem ser limitados por uma boa posição do sensor (em frente do utilizador com um ângulo de elevação no intervalo $[-10^o;10^o]$), assim como com um bom posicionamento do utilizador face ao Kinect (a uma distância mínima de $0.7m$).

Discutindo a eficiência do classificador K-NN quando comparado ao SVM, em tempo real, os resultados são paralelos com os observados acima aquando da análise ao conjunto de amostras de treino utilizado. Ambos os classificadores são capazes de detetar todos os gestos estudados,

embora o classificador SVM demonstre um incidência de erro superior na separação das classes **L** e **T**.

Finalmente, como o *Depth Stream* opera a $30 fps$ a resposta do processamento é rápida e não apresenta atrasos notórios.

5.2.4 Apreciação global

Ao finalizar a análise dos módulos que perfazem este sistema e do seu comportamento como um todo, resta-nos fazer uma apreciação global da qualidade da sua operação.

O sistema atinge efetivamente os objetivos para o qual foi desenhado. Consegue detetar as classes cujos padrões foram usados para treinar os classificadores estudados, demonstrando uma baixa incidência de erros e operando em tempo real.

Como o modelo de deteção não necessita da informação proveniente do *Color Stream*, o seu processamento é mais rápido e o consumo de memória não é tão elevado quanto o do sistema de deteção da expressão facial.

5.3 Sumário

Neste capítulo analisamos os mecanismos utilizados, na implementação da solução desenvolvida, particularmente no que diz respeito ao seu funcionamento e a sua capacidade de satisfazer os objetivos propostos.

Começou-se por estudar as limitações impostas pelo sensor Kinect às aplicações desenvolvidas. A aplicação de deteção da expressão facial é a que mais sofre com estas restrições, seja pela necessidade de usar o modo de captura do sensor RGB em alta resolução, o que diminui a captura para apenas $12 fps$, pela limitação do ângulo de captura ou pela pouca robustez do *Face Tracking SDK* face à morfologia da face do utilizador. Vimos que para uma melhor deteção por parte deste SDK, o utilizador não deverá ter características faciais muito distintas como óculos ou barba forte e carregada.

Outra limitação que é imposta ao sistema de reconhecimento da expressão facial desenvolvido é o número reduzido de unidades de animação disponíveis o que dificulta um perfeito mapeamento de todas as expressões faciais. Estudou-se também o comportamento das unidades de animação em tempo real, chegando à conclusão de que para um melhor desempenho é aconselhável que o utilizador mantenha uma expressão neutra no momento inicial da deteção para minimizar erros.

Chegou-se à conclusão, porém, que as limitações impostas pelo Kinect ao sistema não são impeditivas. O processamento a $12 fps$ continua a ser aceitável dado o tempo que demora a realizar as expressões faciais que o sistema deve detetar e um bom posicionamento do utilizador relativamente ao sensor aumenta o seu grau de liberdade de movimentos. No caso do utilizador ter barba e o sistema não reconhecer os lábios corretamente, uma nova deteção (que pode ser forçada se o utilizador sair do campo de visão do sensor) geralmente é suficiente para corrigir o erro.

Passou-se então à análise mais aprofundada do sistema de reconhecimento de gestos estáticos. Neste caso não há muitas limitações impostas pelo sensor, a única que se notou foi no rastreio

das articulações pelo *Skeleton Stream* que pode resultar em erro, embora este seja momentâneo e facilmente corrigido com o movimento do utilizador, não afetando o sistema em si.

Avançou-se o estudo, passando a avaliar o processo de segmentação implementado. Viu-se que este é adequado ao caso do gesto estático uma vez que consegue efetivamente extrair, da imagem de profundidade, a mão e a sua forma corretamente. Tem, no entanto, a restrição que a mão deve estar isolada no espaço, isto é, a mão não pode estar em contacto com nenhum objeto ou parte do corpo para se obter uma boa segmentação.

Analisando as características implementadas chegou-se à conclusão que a análise do envelope convexo não é suficiente para uma boa deteção devido à elevada probabilidade de gerar informação errónea. Portanto, usaram-se os 7 momentos invariantes de Hu e os ângulos das duas primeiras componentes principais para criar padrões que foram utilizados pelo classificador para detetar os gestos do utilizador. Notou-se que seria necessário normalizar os ângulos utilizados e chegou-se à conclusão que o intervalo $[-0.1; 0.1]$ devolve os resultados mais satisfatórios.

Finalmente, constatou-se que um maior número de amostras/gesto leva a uma melhor viabilidade dos classificadores. Por fim, observou-se que o classificador K-NN, quando $K = 5$, apresenta o melhor desempenho, com a menor taxa de erro.

Capítulo 6

Conclusões

Neste capítulo abordam-se os principais resultados obtidos e as conclusões que se podem retirar do sistema realizado, após o trabalho de desenvolvimento e análise, relativamente à sua capacidade de responder aos objetivos propostos no início desta dissertação.

Termina-se o capítulo com uma discussão de trabalho futuro, na qual se propõem possíveis melhorias aos sistemas de deteção e uma visão mais abrangente das capacidades de um sistema robusto de deteção e interpretação de Língua Gestual.

6.1 Resultados

Os objetivos propostos para esta dissertação passavam pela criação de um sistema de deteção de elementos da Língua Gestual utilizando o sensor Kinect e as ferramentas de desenvolvimento disponibilizadas pela Microsoft. Para o efeito, propusemo-nos a provar que com este sensor é possível criar mecanismos eficazes no reconhecimento da expressão facial e de gestos estáticos.

Começamos pelo reconhecimento da expressão facial. Como vimos nos capítulos anteriores, foi desenvolvida uma aplicação capaz de detetar e reconhecer efetivamente esta vertente da Língua Gestual. O facto das expressões faciais utilizadas no âmbito desta língua durarem o mesmo tempo que leva a realizar um gesto representa um ponto positivo para o sistema, tornando a deteção mais fácil. Utilizando o sistema *Face Tracking SDK* e as unidades de animação disponibilizadas por este provou-se que através de simples parametrização das expressões faciais consegue-se obter reconhecimento destas com boa eficiência e em tempo real. Não se podem descartar, no entanto, as limitações do sistema, como fraca robustez a características da cara do utilizador e o número reduzido de unidades de animação que dificulta uma parametrização de todas as expressões faciais utilizadas na Língua Gestual Portuguesa.

Deve-se referir que o sistema proposto para deteção da expressão facial não contém pontos suficientes para seguir com precisão e estabilidade o movimento das bochechas, uma vez que nenhuma das unidades de animação permite acompanhar variações destas. Assim, através do sistema apresentado não se conseguem parametrizar as expressões "bochecha cheia" e "bochecha vazia". O *Face Tracking SDK* também não consegue detetar a língua dificultando a parametrização

da expressão "língua entre os dentes". As restantes expressões são possíveis de parametrizar através do modelo apresentado.

Neste trabalho apresentou-se, também, um sistema de reconhecimento de gestos estáticos capaz de os detetar através de mecanismos bastantes simples. O processo de deteção da área de interesse da mão utiliza a informação do *Skeleton Stream* reduzindo em muito a complexidade deste processo. A técnica empregue para segmentação, baseada na limiarização das profundidades retorna também bons resultados em tempo real. Todo o processo de pré-processamento é portanto simplificado pelo uso do Kinect.

Para o reconhecimento e deteção de padrões de gestos, o uso de *Emgu CV* permitiu a implementação de algoritmos otimizados para o cálculo de características. Demonstrou-se que o uso dos momentos invariantes de Hu juntamente com os ângulos das duas componentes principais normalizadas retorna bons resultados em tempo real. O sistema consegue efetivamente reconhecer com boa viabilidade todas as classes de gestos com que foi treinado, embora para padrões muito similares por vezes se depare com algumas dificuldades. Por efeito, o classificador K-NN é bastante eficiente. Uma característica associada a este classificador e ao uso dos momentos invariantes da imagem está na capacidade do sistema detetar a mão esquerda, com alguma viabilidade, tendo sido treinado apenas com padrões da mão direita. Finalmente, o uso de vetores de caraterísticas limitados ajuda num processo de classificação mais rápido e permite atingir um bom desempenho.

O sensor Kinect é usado nos dois sistemas de reconhecimento simplificando-os consideravelmente. Prova-se que, com este sensor, consegue-se de facto conceber sistemas simples capazes de reconhecer expressões faciais e gestos estáticos utilizados na Língua Gestual Portuguesa. Damos portanto como cumpridos os objetivos propostos por esta dissertação.

6.2 Trabalho Futuro

Concluída a análise de resultados obtidos nesta dissertação passa-se a referir melhoramentos e extensões ao trabalho desenvolvido.

Em primeiro lugar, temos que considerar o sensor Kinect. A maior resolução a que este consegue operar é de apenas 1280×960 pixels, sendo de apenas metade para a informação de profundidade. Como vimos, esta resolução limita a operação do sistema de reconhecimento da expressão facial. Para a aplicação de reconhecimento de gestos estáticos, dispomos de apenas 640×480 pixels de informação de profundidade, o que resulta numa região de interesse onde se encontra a mão de, no máximo, 250×250 pixels. O número baixo de pixels para descrever a mão do utilizador não permite uma boa descrição da forma, limitando a capacidade de reconhecer alterações muito pequenas na variação da pose.

Perto da finalização do processo de desenvolvimento deste trabalho, a Microsoft apresentou a nova iteração da sua consola de jogos, a *Xbox One* [Xbo13]. Juntamente com a consola será lançada a nova versão do sensor Kinect. Este vem melhorado, agora equipado com uma câmara RGB com resolução 1080p, ou seja 1920×1080 pixels, a $60 fps$. O sistema de infravermelhos deste novo Kinect foi também melhorado, passando a funcionar com tecnologia *time-of-flight*.

Esta tecnologia permitirá obter um detalhe, na informação de profundidade, muito superior ao conseguido com a versão atual do sensor.

O novo sensor trará maior resolução em todos os aspetos que são utilizados no sistema desenvolvido. Assim, fará todo o sentido implementar este utilizando esta nova iteração do Kinect, procurando analisar o efeito que a maior definição disponível tem sobre os produtos desenvolvidos.

Reconhecimento da expressão facial

O sistema de deteção facial desenvolvido pretendeu apenas servir como uma prova do conceito proposto para reconhecimento de expressões faciais. Portanto, foram parametrizadas apenas duas expressões. Propõe-se em primeiro lugar, a modelação e implementação das restantes configurações utilizadas na Língua Gestual Portuguesa.

Como vimos, o reduzido número de unidades de animação disponível não permite a deteção do estado das bochechas do utilizador. A análise da viabilidade de outras técnicas de deteção da face humana, como características de Haar ou a implementação direta do sistema CANDIDE-3, pode vir a provar-se eficiente nesta tarefa.

Uma das fraquezas do sistema desenvolvido, como vimos na secção 5.1.2, está na baixa resistência a características faciais como barba. Utilizando a interface C# do *Face Tracking SDK* não se tem acesso direto às unidades de forma (*Shape Units*) do utilizador. No entanto, estas estão disponíveis através da interface C++ do SDK. Sugere-se então a reconstrução do sistema de deteção da expressão facial utilizando esta interface. O acesso às unidades de forma deverá permitir um melhor controlo e resistência a características faciais, ao desenvolver um modelo da expressão neutra mais dinâmico que o disponível através da interface C#.

Reconhecimento de gestos estáticos

Não está disponível nenhum *dataset* de ortografia gestual da LGP vocacionado para imagens de profundidade. Por isso, teve-se que criar um conjunto de padrões para servirem de conjunto de treino. Devido a questões logísticas e de tempo, este conjunto foi baseado num só indivíduo o que é limitativo. Um conjunto de padrões de treino deve ter grande variabilidade nas amostras. Pelo que deve ser composto por padrões extraídos de vários indivíduos. Outro fator a ter em conta com o conjunto de amostras de treino, quando se usa o classificador K-NN, é que, para este, o número de amostras por gesto não deve ser igual para todas as classes. O classificador K-NN produz melhores resultados quando o número de amostras por gesto, no conjunto de treino, é representativo da frequência de incidência de cada classe. Assim, propõe-se a criação de um *dataset* composto de um número de amostras elevado para cada classe, e extraído a partir de vários indivíduos. Uma análise de incidência de letras é aconselhável por exemplo, as letras “K” e “W” apresentam uma taxa de ocorrência baixa na Língua Portuguesa, podendo então o seu número total de amostras ser inferior. Por outro lado as vogais têm a maior frequência, necessitando de mais amostras por classe.

Neste projeto analisaram-se dois classificadores: K-NN e SVM. Na área de *machine learning* existem outros classificadores como *Hidden Markov Models*, *Random Forests* e redes neuronais artificiais. Propõe-se, também, fazer a análise do sistema quando são utilizados este tipo de algoritmos a fim de procurar o que apresenta melhores resultados.

Finalmente o número de características utilizadas para descrever cada classe é relativamente baixo. O aumento do número de características pode vir a melhorar os resultados de viabilidade obtidos. Recomenda-se, portanto, o estudo de desempenho de descritores de Fourier ao invés dos momentos da imagem, assim como a inclusão de mais características representativas da mão. Considerar características que têm um conhecimento da forma do objeto detetado é também desejável. A análise do envelope convexo do objeto foi abordada com este objetivo mas, como vimos, os seus resultados não eram fidedignos. Propõe-se então o desenvolvimento de um modelo descritivo da mão através das suas partes como o apresentado por Keskin *et al.* [KKKA13]. Este tipo de modelo permitirá obter uma boa descrição da mão e dos dedos, melhorando a deteção dos gestos. Pode-se, então, passar a fazer a deteção de gestos através da posição da mão em vez da análise da forma que toma.

Reconhecimento de gestos dinâmicos

Uma vez que o sensor Kinect consegue seguir o movimento humano com boa viabilidade, os sistemas desenvolvidos neste trabalho procuraram criar métodos que detetassem as restantes vertentes da Língua Gestual Portuguesa. No que toca às características manuais, focámo-nos apenas na ortografia gestual. Acredita-se ser possível usar este sistema como base para um sistema de deteção de gestos dinâmicos. Como um gesto dinâmico se propaga durante o tempo, pode-se criar um sistema de decisão, para cada gesto, que determina as suas partes ao longo do tempo. Desta forma, o sistema seria capaz de decidir que gesto está a ser desenhado a partir das suas partes.

Propõe-se portanto um sistema de deteção de gestos composto por duas camadas: a primeira coincide com o sistema de deteção de gestos estáticos apresentado, a segunda camada consiste num modelo de decisão que passa por partir cada gesto dinâmico em vários segmentos e identificar cada um destes.

Sistema de interpretação da Língua Gestual Portuguesa

Provámos com o trabalho apresentado que o sensor Kinect é capaz de analisar e reconhecer todas as vertentes das Língua Gestual: o acompanhamento do movimento dos braços e posição do corpo através da informação disponibilizada no *Skeleton Stream*; a expressão facial através do *Face Tracking SDK*; elementos manuais através da aplicação de métodos de visão por computador e *Machine Learning*, onde a informação de profundidade produzida pelo sensor permite reduzir a complexidade do processo de deteção e segmentação.

Consideramos ser possível utilizar o sensor Kinect e estas técnicas para criar um sistema capaz de interpretar a Língua Gestual Portuguesa. A junção do reconhecimento do movimento do

utilizador, de ortografia gestual, de expressão facial e de gestos dinâmicos não é uma tarefa demasiado complexa uma vez que todos os sinais são extraídos da mesma fonte. Ao utilizar técnicas de análise linguísticas, atualmente disponíveis para sistemas de reconhecimento de fala, pode-se criar um modelo que use a glosa (um tipo de texto escrito que ajuda na tradução de uma língua falada para uma língua gestual) como uma interface entre a articulação do gesto e a interpretação final.

Propõe-se começar por um sistema que seja capaz de pegar em texto descrito em glosa e modelar o respetivo texto em Língua Gestual, passando a utilizar esta informação para treinar o sistema. Uma vez tendo um sistema capaz de interpretar texto para gesto, consegue-se realizar o processo inverso, criando assim um sistema de interpretação bidirecional da Língua Gestual Portuguesa.

Referências

[Ahl01] Jörgen Ahlberg. Candide-3-an updated parameterised face. 2001.

[ArG12] Sigma ArGe. SigmaNIL, 2012. Último acesso em 2012/02/06. URL: http://www.sigmanil.com/.

[AS03] V. Athitsos e S. Sclaroff. Estimating 3D hand pose from a cluttered image. volume vol.2, pages 432 – 9, Los Alamitos, CA, USA, 2003.

[Bal10] A.B. Baltazar. *Dicionário de Língua Gestual Portuguesa*. Porto Ed., 2010.

[Biz12] Microsft BizSpark. The Microsoft Acelebrator for Kinect, 2012. Último acesso em 2012/02/06. URL: http://www.microsoft.com/bizspark/kinectaccelerator/.

[BK08] Gary Bradski e Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Incorporated, 2008.

[Bre12] Jasper Brekelmans. Brekel Kinect, 2012. Último acesso em 2012/02/06. URL: http://www.brekel.com/.

[CGZZ10] Qin Cai, David Gallup, Cha Zhang e Zhengyou Zhang. 3D deformable face tracking with a commodity depth camera. volume 6313 LNCS, pages 229 – 242, Heraklion, Crete, Greece, 2010.

[CLV12] L. Cruz, D. Lucio e L. Velho. Kinect and rgbd images: challenges and applications. pages 36 – 49, Los Alamitos, CA, USA, 2012.

[CV12] Emgu CV. Emgu CV : OpenCV in .NET, 2012. Último acesso em 2013/6/12. URL: http://www.emgu.com/.

[Dar71] C.R. Darwin. *The descent of man, and selection in relation to sex*. The descent of man, and selection in relation to sex. 1871.

[Dav96] ER Davies. *Machine vision: Theory, algorithms, practicalities*. Academic Press, London, 1996.

[dS11] Associação Portuguesa de Surdos. Alfabeto manual, 2011. Último acesso em 2012/02/06. URL: http://www.apsurdos.org.pt/.

[DSM+11] P. Doliotis, A. Stefan, C. McMurrough, D. Eckhard e V. Athitsos. Comparing gesture recognition accuracy using color and depth information. In *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments*, page 20. ACM, 2011.

[DT05] N. Dalal e B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886 –893 vol. 1, june 2005. doi:10.1109/CVPR.2005.177.

[EDHD02] Ahmed Elgammal, Ramani Duraiswami, David Harwood e Larry S. Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of the IEEE*, 90(7):1151 – 1162, 2002.

[EF77] Paul Ekman e Wallace V Friesen. Facial action coding system. *Consulting Psychologists Press, Stanford University, Palo Alto*, 1977.

[EHD00] A. Elgammal, D. Harwood e L. Davis. Non-parametric model for background subtraction. *Computer Vision—ECCV 2000*, pages 751–767, 2000.

[Ekm99] P. Ekman. Basic emotions, T. Dalgleish, MJ Power, Editors. *Handbook of cognition and emotion*, pages 45–60, 1999.

[ELSvG08] A. Ess, B. Leibe, K. Schindler e L. van Gool. A mobile vision system for robust multi-person tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1 –8, june 2008. doi:10.1109/CVPR.2008.4587581.

[FGMR10] P.F. Felzenszwalb, R.B. Girshick, D. McAllester e D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627 –1645, sept. 2010. doi:10.1109/TPAMI.2009.167.

[FH01] P.F. Felzenszwalb e D.P. Huttenlocher. Pictorial structures for object recognition. *Int. J. Comput. Vis. (Netherlands)*, 61(1):55 – 79, 2005/01/.

[FL06] Kikuo Fujimura e Xia Liu. Sign recognition using depth image streams. volume 2006, pages 381 – 386, Southampton, United kingdom, 2006.

[FR97] N. Friedman e S. Russell. Image segmentation in video sequences: A probabilistic approach. In *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*, pages 175–181. Morgan Kaufmann Publishers Inc., 1997.

[FTR+04] R. Feris, M. Turk, R. Raskar, K. Tan e G. Ohashi. Exploiting depth discontinuities for vision-based fingerspelling recognition. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on*, pages 155–155. IEEE, 2004.

[fWT12] Kinect for Windows Team. Near mode: What it is (and isn't), Janeiro 2012. Último acesso em 2012/02/05. URL: http://blogs.msdn.com/b/kinectforwindows/archive/2012/01/20/near-mode-what-it-is-and-isn-t.aspx.

[GBCR00] Xiang Gao, T.E. Boult, F. Coetzee e V. Ramesh. Error analysis of background adaption. volume vol.1, pages 503 – 10, Los Alamitos, CA, USA, 2000.

[GSRL98] W.E.L. Grimson, C. Stauffer, R. Romano e L. Lee. Using adaptive tracking to classify and monitor activities in a site. pages 22 – 9, Los Alamitos, CA, USA, 1998//.

[GW02] Rafael C Gonzales e Richard E Woods. *Digital Image Processing, 2-nd Edition*. Prentice Hall, 2002.

[HAS03] J. Han, G. Awad e A. Sutherland. Automatic skin segmentation and tracking in sign language recognition. *IET Comput. Vis. (UK)*, 3(1):24 – 35, 2009/03/.

[Hjo69] Carl-Herman Hjortsjö. Människans ansikte och det mimiska språket. *Studentliteratur, Lund, Sweden*, 1969.

[HSL07] Seok-Ju Hong, Nurul Arif Setiawan e Chil-Woo Lee. Real-time vision based gesture recognition for human-robot interaction. volume 4692 LNAI, pages 493 – 500, Vietri sul Mare, Italy, 2007.

[HT+85] Kazuhiro Homma, Ei-ichi Takenaka et al. An image processing method for feature extraction of space-occupying lesions. *Journal of nuclear medicine: official publication, Society of Nuclear Medicine*, 26(12):1472, 1985.

[Hu62] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2):179–187, 1962.

[ILI98] K. Imagawa, Shan Lu e S. Igi. Color-based hands tracking system for sign language recognition. pages 462 – 7, Los Alamitos, CA, USA, 1998.

[Inc12] Jintronix Inc. JINTRONIX, 2012. Último acesso em 2012/02/06. URL: http://www.jintronix.com/.

[ISO99] ISO. *ISO/IEC 14496-2:1999: Information technology — Coding of audio-visual objects — Part 2: Visual*. 1999. Available in English only. URL: http://www.iso.ch/cate/d25034.html.

[Its13] Itseez. OpenCV, Open Source Computer Vision, 2013. Último acesso em 2013/6/12. URL: http://opencv.org/.

[Jen99] C. Jennings. Robust finger tracking with multiple cameras. pages 152 – 60, Los Alamitos, CA, USA, 1999.

[KE96] Mohammed Waleed Kadous e Computer Science Engineering. Machine recognition of auslan signs using powergloves: Towards large-lexicon recognition of sign language. In *Proceedings of the Workshop on the Integration of Gesture in Language and Speech*, pages 165–174, 1996.

[Kin11] KinectHacks.net. KinectHacks.net, 2011. Último acesso em 2012/02/06. URL: http://www.kinecthacks.com/.

[KKKA13] Cem Keskin, Furkan Kıraç, YunusEmre Kara e Lale Akarun. Real time hand pose estimation using depth sensors. In Andrea Fossati, Juergen Gall, Helmut Grabner, Xiaofeng Ren e Kurt Konolige, editors, *Consumer Depth Cameras for Computer Vision*, Advances in Computer Vision and Pattern Recognition, pages 119–137. Springer London, 2013. URL: http://dx.doi.org/10.1007/978-1-4471-4640-7_7.

[KLF12] Xiaofeng Ren Kevin Lai, Liefeng Bo e Dieter Fox. RGB-D Object Dataset, Novembro 2012. Último acesso em 2012/02/06. URL: http://www.cs.washington.edu/rgbd-dataset/.

[MHK11] T.B. Moeslund, A. Hilton e V. Kruger. A survey of advances in vision-based human motion capture and analysis. *Comput. Vis. Image Underst. (USA)*, 104(2-3):90 – 126, 2006/11/.

[MHKS11] Thomas B. Moeslund, Adrian Hilton, Volker Krüger e Leonid Sigal, editors. *Visual Analysis of Humans - Looking at People*. Springer, 2011.

[Mic12] Microsoft. MSDN - Kinect for Windows Sensor, 2012. Último acesso em 2012/02/05. URL: http://msdn.microsoft.com/en-us/library/hh855355.aspx.

[MP04] A. Mittal e N. Paragios. Motion-based background subtraction using adaptive kernel density estimation. volume Vol.2, pages 302 – 9, Los Alamitos, CA, USA, 2004//.

[MSD12] MSDN. Kinect Face Tracking, 2012. Último acesso em 2012/06/06. URL: http://msdn.microsoft.com/en-us/library/jj130970.aspx.

[MSMCMCCP01] R. Munoz-Salinas, R. Medina-Carnicer, F.J. Madrid-Cuevas e A. Carmona-Poyato. Depth silhouettes for gesture recognition. *Pattern Recognit. Lett. (Netherlands)*, 29(3):319 – 29, 2008/02/01.

[NSF12] Pushmeet Kohli Nathan Silberman, Derek Hoiem e Rob Fergus. NYU Depth Dataset V2, 2012. Último acesso em 2012/02/06. URL: http://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html.

[OKA11] I. Oikonomidis, N. Kyriazis e A. Argyros. Efficient model-based 3D tracking of hand articulations using Kinect. *BMVC, Aug*, 2, 2011.

[PB11] N. Pugeault e R. Bowden. Spelling it out: Real-time asl fingerspelling recognition. pages 1114 – 19, Piscataway, NJ, USA, 2011.

[Pin07] S. Pinker. *The Language Instinct: How the Mind Creates Language*. P. S. Series. HarperCollins, 2007.

[RVRD08] A. Rezaei, M. Vafadoost, S. Rezaei e A. Daliri. 3D pose estimation via elliptical Fourier descriptors for deformable hand representations. pages 1871 – 5, Piscataway, NJ, USA, 2008.

[Ryd87] M Rydfalk. Candide: A parameterized face, linkoping university, linkoping, sweden, rep. *LiTH-ISY-I-0866*, 1987.

[Sax09] Ashutosh Saxena. Cornell RGBD Dataset, 2009. Último acesso em 2012/02/06. URL: http://pr.cs.cornell.edu/sceneunderstanding/data/data.php.

[SFC+11] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman e A. Blake. Real-time human pose recognition in parts from single depth images. pages 1297 – 304, Piscataway, NJ, USA, 2011.

[SK99] J. Segen e S. Kumar. Shadow gestures: 3D hand pose estimation using a single camera. volume Vol. 1, pages 479 – 85, Los Alamitos, CA, USA, 1999.

[SMO03] J. Stander, R. Mech e J. Ostermann. Detection of moving cast shadows for object segmentation. *IEEE Trans. Multimed. (USA)*, 1(1):65 – 76, 1999/03/.

[SWP98] Thad Starner, Joshua Weaver e Alex Pentland. Real-time american sign language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1371 – 1375, 1998.

[TKBM99] K. Toyama, J. Krumm, B. Brumitt e B. Meyers. Wallflower: principles and practice of background maintenance. volume vol.1, pages 255 – 61, Los Alamitos, CA, USA, 1999//.

[UI12] inc. Ubi Interactive. rbi - Turns every surface into a 3D multitouch screen, 2012. Último acesso em 2012/02/06. URL: http://www.ubi-interactive.com//.

[VG08] Christian Vogler e Siome Goldenstein. Facial movement analysis in ASL. *Universal Access in the Information Society*, 6(4):363 – 374, 2008.

[VM04] C. Vogler e D. Metaxas. Handshapes and movements: Multiple-channel american sign language recognition. *Gesture-Based Communication in Human-Computer Interaction*, pages 431–432, 2004.

[WA12] J. Webb e J. Ashley. *Beginning Kinect Programming with the Microsoft Kinect SDK*. Apress, 2012.

[WADP96] Christopher R. Wren, Ali J. Azarbayejani, Trevor J. Darrell e Alexander P. Pentland. Pfinder: real-time tracking of the human body. volume 2615, pages 89 – 98, Philadelphia, PA, USA, 1996.

[Wel91] Bill Welsh. *Model-based coding of images.* PhD thesis, University of Essex, 1991.

[WN06] Bo Wu e Ram Nevatia. Tracking of multiple, partially occluded humans based on static body part detection. volume 1, pages 951 – 958, New York, NY, United states, 2006.

[WTK12] Robert Wang, Chris Twigg e Kenrick Kin. 3Gear Systems, 2012. Último acesso em 2012/02/06. URL: http://threegear.com/.

[Xbo13] Xbox. Introducing Xbox One, 2013. Último acesso em 2013/6/21. URL: http://www.xbox.com/en-US/xboxone/meet-xbox-one.

[YDr12] YDreams. YScope : YDreams and Hospital Santa Maria da Feira Launch Revolutionary App in the Health and Technology Sectors, Julho 2012. Último acesso em 2012/02/04. URL: http://www.ydreams.com/index.php#/en/aboutus/media/whatsup/2012/YSCOPEYDREAMSHOSPITALSANTAMARIA/.

[ZCF+04] L.G. Zhang, Y. Chen, G. Fang, X. Chen e W. Gao. A vision-based sign language recognition system using tied-mixture density HMM. In *International Conference on Multimodal Interfaces: Proceedings of the 6 th international conference on Multimodal interfaces*, volume 13, pages 198–204, 2004.

[Zha02] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE Multimed. (USA)*, 19(2):4 – 10, 2012/02/.

[ZK04] J. Zieren e K.F. Kraiss. Non-intrusive sign language recognition for human-computer interaction. In *Proc. IFAC/IFIP/IFORS/IEA symposium on analysis, design and evaluation of human machine systems*, 2004.